AD-A071 075 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 17/7

A DEVELOPMENTAL COMPUTER MODEL FOR INVESTIGATIONS OF AIR TRAFFI--ETC(U)

MAR 79 J T MALOKAS, A P PEDERSON

UNCLASSIFIED NL

1 OF 2

AD

A071075

2 LEVEL II

AD A071075

NAVAL POSTGRADUATE SCHOOL

Monterey, California

DDC
RECEIVED
JUL 12 1979
B

9 Master's THESIS

6 A DEVELOPMENTAL COMPUTER MODEL FOR INVESTIGATIONS OF AIR TRAFFIC MANAGEMENT PROBLEMS: A CASE INVESTIGATING TWO DECISION STRATEGIES.

by

10 John Thomas/Malokas, Jr.

~~and~~

Arvid Paul/Pederson

12 133p.

11 March 1979

Thesis Advisor: E. F. Roland

Approved for public release; distribution unlimited

251 450

79 07 11 050

DDC FILE COPY

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Developmental Computer Model for Investigations of Air Traffic Management Problems: A Case Investigating Two Decision Strategies		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John Thomas Malokas, Jr. Arvid Paul Pederson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1979
		13. NUMBER OF PAGES 132
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Air Traffic Management
Air Traffic Control
Regional Scheduling
Air Traffic Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computer simulation model designed to help solve regional air traffic scheduling problems was developed. Bases, mission areas, and aircraft were modeled using the simulation language SIMSCRIPT. Events in the simulation included takeoffs, departures, enroutes, missions, arrivals at Initial Approach Fixes (IAFs) and landings.

To demonstrate potential use of the model, the problem of rescheduling Strategic Air Command (SAC) aircraft upon

one

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

base closures was addressed. Two strategies for the diversion of such aircraft were developed, implemented and the results compared on the basis of impact on final destination bases and average aircraft airborne time. Strategy 1 entailed the rerouting of aircraft to designated alternate bases or to the nearest open base without constraint. Strategy 2 involved the selection of an alternate base by insuring that parking spaces and appropriate maintenance support were available.

Finally, extensions to the model and recommendations are discussed.

A Developmental Computer Model for Investigations of Air Traffic Management Problems: A Case Investigating Two Decision Strategies

by

John Thomas Malokas, Jr.
Major, United States Air Force
B.S., Ohio University, 1966
M.S., University of Southern California, 1973

and

Arvid Paul Pederson
Captain, United States Air Force
B.S., University of North Dakota, 1967

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONAL DECISION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1979

Author: John T. Malokas Jr.

Author: Arvid Paul Pederson

Approved by: Ellen F. Boland, Thesis Advisor

Alvin F. Andrus, Co-Advisor

John M Wozencraft, Chairman, C3 Academic Group

Academic Dean

ABSTRACT

A computer simulation model designed to help solve regional air traffic scheduling problems was developed. Bases, mission areas, and aircraft were modeled using the simulation language SIMSCRIPT. Events in the simulation included takeoffs, departures, enroutes, missions, arrivals at Initial Approach Fixes (IAFs) and landings.

To demonstrate potential use of the model, the problem of rescheduling Strategic Air Command (SAC) aircraft upon base closures was addressed. Two strategies for the diversion of such aircraft were developed, implemented and the results compared on the basis of impact on final destination bases and average aircraft airborne time. Strategy 1 entailed the re-routing of aircraft to designated alternate bases or to the nearest open base without constraint. Strategy 2 involved the selection of an alternate base by insuring that parking spaces and appropriate maintenance support were available.

Finally, extensions to the model and recommendations are discussed.

TABLE OF CONTENTS

I. THE PROBLEM

A. INTRODUCTION

Air traffic management has been a controversial topic in recent years but several tragic accidents in the past two years have refocused concern on the need for safer aircraft operations. The mid-air collision in September of 1978 involving a commercial Boeing 727 and a single-engine Cessna aircraft, called the worst disaster in the history of aviation in the United States, highlighted the need for better air traffic management. Because a military terminal radar facility exercised control of the commercial aircraft and transferred control to a civilian airport tower minutes before the mid-air collision, the need for Department of Defense (DOD) involvement in the improvement of air traffic management became more evident.

In mid-1975, a Headquarters, United States Air Force (HQ, USAF) General Officers' Panel was convened to investigate mid-air collision problems. An initiative which resulted from this investigation was the Air Training Command's SMOOTH FLOW Program. This provided guidelines for the scheduling of air operations to preclude air traffic saturation at command bases. The investigation revealed that at least four factors bear on the high collision potential and air traffic saturation problem in terminal areas: (1) increasing number of military air operations, (2) increasing number of

civil air operations, (3) increasing consolidation and restriction on use of airspace, and (4) increasingly complex scheduling problems. The investigation further revealed that readiness and training requirements precluded major decreases in military air operations. Since the second and third factors were beyond the direct control of DOD, the fourth factor, scheduling, was the area where improvements could be made. Suggestions offered by the study to ameliorate conditions detrimental to smooth and orderly terminal air traffic included integrated base and regional scheduling methods.

The subject of this thesis is the rescheduling of aircraft to alternate bases when base closures occur. These base closures sometimes occur with little or no warning such as when an aircraft crashes or experiences tire failures on takeoff or landing. Probably the most common reason for base closure is weather phenomena which preclude runway operations and often affect several bases at the same time. The absorption of the air traffic from the closed bases into the terminal areas of alternate bases greatly complicates the air traffic control situation. Additional considerations such as low fuel states, weather, crewmember experience levels, maintenance support capability and security requirements confront the military commanders and planners and require decisive actions in order to recover all aircraft safely.

B. *BACKGROUND*

The investigation into the feasibility of developing a regional scheduling scheme involved the analysis of air operations and their attendant unit scheduling methods. This investigation revealed some interesting factors concerning air traffic management. First, and not surprisingly, the orderly flow of traffic at an air base is highly dependent on the numbers, missions, and types of assigned and transient aircraft. For example, the air traffic control situation at an Air Force pilot training base with its high density of similar aircraft is very different from that at an air base serving all sizes of jet and propeller-driven aircraft including helicopters.

A second factor in air traffic management is that a certain level of integrated base and regional scheduling exists within the Air Force and other services but it is often a fragmented process which has resulted because of conflicts, mission requirements, or joint use requirements and not because of a well-conceived grand strategy. Examples of this coordination or scheduling integration include air refueling, fighter intercept training, bombing and gunnery range practice, and joint service exercises. Often these activities are governed by inter-governmental, inter-service, inter-command, or joint use agreements. The pervasiveness and complexity of these agreements greatly complicate attempts by any one service or agency to provide better air traffic management.

A third factor is that the Air Traffic Control (ATC) facility at an air base operates under Federal Aviation Regulations (FAR's) and other national and local regulations and agreements which are not under the operational control of local or regional commanders. Certainly the commanders influence the traffic controls at their installations but they do not directly control them under normal conditions. Even special interest and congested bases which are classified as Prior Permission Required (PPR) in the Instrument Flight Rules (IFR) Enroute Supplement to the DOD Flight Information Publications (FLIP) cannot deny terminal area privileges to emergency aircraft. The DOD FLIP embodies the policies and procedures which apply to military operations (over and above applicable civilian regulations) and constitutes a higher authority than the operational commander.

The fourth factor concerns the very complexity of scheduling a single unit's operational and training requirements. Interviews with unit schedulers have revealed that constraints involving maintenance requirements, environmental considerations, collateral operational or training commitments and others severely limit the flexibility required in making large-scale changes in unit scheduling practices. To extend this to basewide or regional levels requires strategies which accommodate these operational and training requirements while simultaneously satisfying

the aforementioned constraints and the objective of smooth and orderly air traffic flow.

Discussions with commanders, air traffic control specialists, aircrewmembers and schedulers have indicated that each regards the terminal air traffic control situation as manageable and unstressed when operations develop as they were planned. However, these same people concede that weather, accidents, exercises and other factors which are not under the control of planners do in fact stress the situation and cause numerous problems. Obviously, planners have developed plans for contengencies but often these plans are in conflict with each other when viewed from a larger perspective. The classic example is the diversion of all aircraft from one base which has a closed runway to the closest open base. Often this results in a stressed condition at the second base which could have been avoided.

Because of the complexity involved in formulating a regional scheduling scheme, it proved beneficial to investigate a subset of all air traffic operations in a region in order to reduce overall regional saturation. A good candidate for this investigation was the air traffic at Strategic Air Command (SAC) bases. The very nature of the SAC aircraft mission was amenable to this regional investigation because the operational command and control (C^2) of the aircraft operations is regional and involves numerous aircraft and bases which are strictly and systematically

controlled. If rescheduling of SAC aircraft to alternate bases is necessary, it is normally accomplished at the regional level with the landing base being another SAC base.

In the continental United States (CONUS), SAC is divided into two numbered air forces (NAFs). Fifteenth Air Force controls most of the SAC bases west of the Mississippi River; and Eighth Air Force controls those bases east of the Mississippi River. The strong centralized control of SAC aircraft forces flows from the tactical wing which normally consists of two or more squadrons to the NAF or intermediate headquarters and than to SAC Headquarters at Offutt Air Force Base (AFB) near Omaha, Nebraska. This flow of control is a subtle one but for peacetime conditions the NAF directs the day-to-day aircraft operations at its bases. This NAF or regional operation was investigated in this thesis.

The locations of SAC bases in the CONUS are depicted at Table I. Each base is identified by a three-letter code corresponding to four-letter International Civil Aviation Organization (ICAO) code. These codes uniquely identify all air bases in the world. In the CONUS the first letter is K so the last three letters also uniquely identify each base of interest. The letters represent navigation facilities at or near bases and are usually associated with cities or geographical features. For

TABLE I

BASE NAME	IDENTIFIER	NEARBY CITY	NAF
Fairchild AFB	KSKA	Spokane, WN	15
Beale AFB	KBAB	Marysville, CA	15
Mather AFB	KMHR	Sacramento, CA	15
Travis AFB	KSUU	Fairfield, CA	15
Castle AFB	KMER	Merced, CA	15
March AFB	KRIV	Riverside, CA	15
Ellsworth AFB	KRCA	Rapid City, SD	15
Minot AFB	KMIB	Minot, ND	15
Grand Forks AFB	KRDR	Grand Forks, ND	15
Offutt AFB	KOFF	Omaha, NE	15
Dyess AFB	KDYS	Abilene, TX	8
McConnell AFB	KIAB	Wichita, KS	8
Altus AFB	KLTS	Altus, OK	8
Carswell AFB	KFWH	Fort Worth, TX	8

example, KRIV or RIV for March AFB, California, represents the name of the navigation facility at the base and also the nearby city of Riverside. Table I contains the names, identifiers, nearby cities, and numbered air forces for SAC bases of interest west of the Mississippi River.

In Chapter II, the computer simulation model is discussed in greater detail. Next, inputs to the program, outputs of the program and decision rules concerning diversion of aircraft when a runway closure occurs are discussed. Also, results of the comparison of two different decision strategies for this diversion situation are discussed. Finally, in Chapter IV limitations of the model, extensions to the model and other recommendations are presented.

II. MODEL DESCRIPTION

A. MODEL SETTING

The environment which was modeled was a military regional air traffic situation, specifically the western region of the Strategic Air Command of the USAF within the continental United States.

Elements of the model included bases, mission areas and aircraft. The bases were 14 Air Force bases which have SAC flying units assigned with each base consisting of a runway and an air traffic pattern. Mission areas were either air refueling tracks, low level routes or "delay" areas with "delay" areas being viewed as holding points, intercept training areas or navigation legs. The mission areas are represented in the model as a single point. Conceptually, the aircraft is scheduled to that point, remains there for the duration of that phase of the mission, then proceeds to the next portion of the mission.

B. PROCESSING ENVIRONMENT

The model was written in the simulation language SIMSCRIPT (version II.5) and was run on an IBM 360/65 computer at the Naval Postgraduate School, Monterey, California. Input was made in the OS/batch mode. Specific formats for the inputs are discussed in Chapter III.

C. DESCRIPTION

1. Overview

The general flow through the model for a typical, single aircraft is as follows. A takeoff time is scheduled for the aircraft at a specific base. At that time, if there is no conflict with landing aircraft at that base, the aircraft takes off and proceeds to the base departure point. There the aircraft goes enroute to a mission area and performs the scheduled activity for that area. Upon completion of the activity the aircraft proceeds to another mission area or returns to the Initial Approach Fix (IAF) of its base. At the IAF it enters a transition phase where it attempts to complete scheduled transition activity such as radar approaches, visual approaches and "touch and go" landings. Upon completion of this activity or when mission time is reached the aircraft makes a full stop landing.

2. Concepts

a. Base Traffic Pattern

The local pattern for the individual bases is depicted in Figure 1. The features of interest are the IAF, the departure, the landing, the overhead pattern, the rectangular pattern and the radar pattern. The IAF is the point of entry to the controlled airspace for the base. It may be physically outside of the controlled airspace for the base. It serves as a navigation point and control point for timing and routing to the local pattern. In this

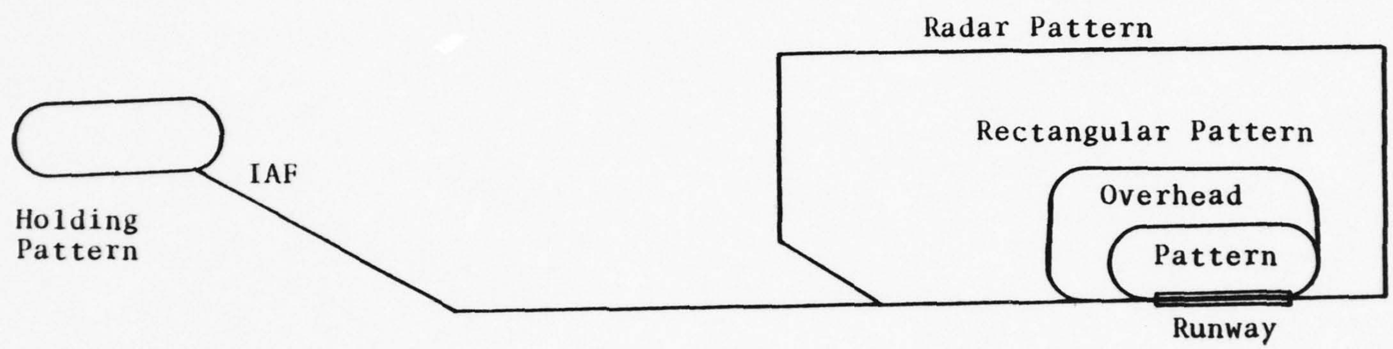


Figure 1. Local Traffic Pattern

simulation it serves as the crossover point to the transition phase of the mission. The departure point is used to make decisions as to which activity will be scheduled next for the aircraft. This would include remaining in the local pattern or going enroute to another base or a particular mission area.

The patterns of the local airspace are the overhead, the rectangular and the radar. The radar pattern may be used by all aircraft and simulates an approach made under control of the radar approach control facility (RAPCON) of the base. The overhead and rectangular patterns are visual patterns. The overhead is used by fighter type aircraft whereas the rectangular pattern is used by heavier aircraft such as B-52s or KC-135s.

The landing point is used in the simulation as the decision point as to who will gain access to the runway. It provides the timing and spacing function for the aircraft coming from the IAF out of the various patterns. Assumptions which were made about the pattern: (1) aircraft in each pattern fly at the same speed while in that pattern. (2) Separation between landings is at least one minute. (3) The distance or time between the landing decision point and the departure point is one minute. (4) The distance between the takeoff point and the departure point is one minute.

b. Time

SIMSCRIPT is an event step simulation language. It maintains an internal clock and a chronologically ordered event list. When the next event comes up in the event list the master clock (represented by the system variable TIME.V) is moved forward to coincide with the occurrence of that event. This master time is maintained in decimal days. Hence, a frequently encountered conversion factor in the program is 1440 minutes/day. In this simulation the master time represents Greenwich Mean Time (GMT) or "Zulu" time.

c. Distance

To facilitate making various computations, all distances are referenced by time in the simulation. Distance, when needed, is Euclidean distance assuming a flat earth. A standard x,y rectangular coordinate system was placed with the origin at 3000 N latitude, 12500 W longitude. This placement located all bases and mission areas of the model in the first quadrant of the x-y plane. The x and y coordinates of fixed points such as bases were then determined by assuming a 400 nautical mile per hour cruise speed for all aircraft. Although the normal cruise speed is higher this speed was selected to compensate for portions of the mission flown at slower speeds such as landing or holding.

3. Detailed Description of the Program

a. PREAMBLE

In the preamble certain initial program conditions are established. First, the background mode is set to real. This means that any unspecified variables introduced in the program are treated as real variables. DEFINE statements are then used to improve the overall readability of the program. Next, the simulation elements along with their attributes are defined. These are:

(1) Mission Areas

X.COOR x-coordinate (in minutes) for the area.

Y.COOR y-coordinate (in minutes) for the area.

(2) Bases

NAME Three-letter ICAO designator for the base.

X.POS x-coordinate (in minutes) for the base.

Y.POS y-coordinate (in minutes) for the base.

TRANSIENT.PARKING.SPACES The number of parking spaces currently available at the base.

TTL.NO.TRANSIENTS The number of transient aircraft which have landed at the base.

AVAIL.NO.TPS The maximum number of parking spaces available at the base.

SEARCH.FLAG A device used in checking certain constraints at a base. If the base fails the check the SEARCH.FLAG value is set to one. Otherwise it is zero.

1.MAINT.SUP.CAP   Either B for bomber (B-52), K for tanker
2.MAINT.SUP.CAP   (KC-135), O for other, or X for none
3.MAINT.SUP.CAP   available.

RUNWAY   Either idle, busy or closed.

MAXIRADAR.QUEUE   Maximum number of aircraft which can
MAXIRECTANGULAR.QUEUE   be in the pattern at any one time
MAXIOVERHEAD.QUEUE

TTL.RECTANGULAR   Total number of occurrences of the activity
TTL.RADAR   at the base during the simulation.

.

.

.

TTL.MISSED.APPROACHES

NO.TAKEOFF.CONFLICTS   Total number of takeoff conflicts.

NO.LANDING.CONFLICTS   Total number of landing conflicts.

(3) Aircraft

TYPE   Either fighter-type or multi-engine.

DESIGN   B for Bomber
K for KC-135
F for fighter
T for trainer
O for other

TAIL.NUMBER   4-digit number.

LOCAL.TIME   The local time in hours and minutes for the aircraft referenced to the home base of the aircraft.

A.HOME.BASE Three-letter ICAO designator for the home base of the aircraft.

HOME.BASE 1-14

A.CURRENT.BASE Three-letter designator for the base at which the aircraft is located.

CRNT.BASE 1-14

A.DESTINATION.BASE Three-letter designator for the base to which the aircraft is enroute.

DESTINATION.BASE 1-14

A.ALTERNATE.BASE Three-letter designator for the alternate base for the aircraft.

ALTERNATE.BASE 1-14

MISSION.TIME.REMAINING The amount of scheduled mission time remaining. Mission time is the total time from takeoff to full stop landing.

EMER.DIVERT.FLAG Either EMERGENCY or DIVERT.

LKT The last known time for the aircraft.

X.LAST.POS The last position for the aircraft. This

Y.LAST.POS position along with the last known time are used to compute current location.

TRANSITION.TIME.REMAINING The amount of time remaining in the transition phase of the mission.

TOTAL.FUEL.REMAINING.TIME The maximum amount of flying time remaining without emergency refueling.

IAF.APPROACHES	The number of occurrences for each activity
MISSED.APPROACHES	which remains to be accomplished.
RADAR.APPROACHES	
VISUAL.APPROACHES	
T.G.LANDINGS	
SCH.IAF.APPROACHES	The scheduled amount of transition activity.
.	
.	
.	
SCH.T.G.LANDINGS	
A.IAFS	The amount of transition activity accomplished.
.	
.	
.	
A.T.GS	
A.1.MSN.AREA	Alpha-numeric designators for the three mission
A.2.MSN.AREA	areas to which an aircraft can be scheduled.
A.3.MSN.AREA	
MSN.1.AREA	1-30
MSN.2.AREA	1-30
MSN.3.AREA	1-30 .
M.2.AREA.TIME	The area entrance times for the aircraft.
M.2.AREA.TIME	
M.3.AREA.TIME	

1.RCVR.TNKR.TAIL.NO 2.RCVR.TNKR.TAIL.NO 3.RCVR.TNKR.TAIL.NO	The tail number for the plane to which the aircraft is paired for an air re-fueling in the particular mission area.
1.ONLOAD.OFFLOAD 2.ONLOAD.OFFLOAD 3.ONLOAD.OFFLOAD	The scheduled onload or offload of fuel (in minutes) in the mission area.
1.MSN.OPTION 2.MSN.OPTION 3.MSN.OPTION	1 = Air refueling 2 = Low level 3 = Delay
1.DURATION 2.DURATION 3.DURATION	The time in minutes for the scheduled activity in the mission area.
LAST.MSN.AREA	The mission area the aircraft was in last.
Z.TAKEOFF Z.IAF Z.LANDING	The time (GMT) in hours and minutes of the occurrence of the event.
L.TO L.IAF L.LNDG	The local time referenced to the aircraft's home base in hours and minutes.
GMT	The current GMT for the aircraft in hours and minutes.
1.ENTRY 2.ENTRY 3.ENTRY	The mission area entrance times for the aircraft in GMT (hours and minutes).
1.EXIT 2.EXIT 3.EXIT	The mission area exit times for the aircraft in GMT (hours and minutes).

1.R.T.DESIGN The design of the plane to which the aircraft
2.R.T.DESIGN is paired for refueling.
3.R.T.DESIGN

MSN.STAGE Either JUST.TOOKOFF
MSN.COMPLETE
OTHER.BASE.PATTERN or
HOMEBASE.PATTERN

MSN.TYPE The overall mission profile for the aircraft.

1: Home base to mission area(s) to home base.
2: Home base to mission area(s) to a transition base to home base.
3: Home base to destination base.
4: Home base to a transition base to home base.
5: Home base to mission area(s) to a destination base (other than home base).

TIM1.IN.TAKQUEUE The time of entrance or total time last
TIME2.IN.OVRQUEUE spent in the queue.
TIM3.IN.RECQUEUE
TIM4.IN.RADQUEUE
TIM5.IN.IAFQUEUE

In addition to these attributes the permanent entities, mission area and base, own sets into which aircraft may be filed. For mission areas there are air refueling tracks, low-level routes or delay areas. For bases these are the takeoff, overhead, rectangular, radar and IAF queues. Events in the simulation are TAKEOFF, DEPARTURE.POINT, ENROUTE, .MISSION, ARRIVAL.AT.IAF,

LANDING, RELEASE,RUNWAY, HALF.HOUR.STATISTICS, STOP.SIMULATION and CHANGE.RUNWAY.STATUS. Each of these will be discussed separately. If a RELEASE.RUNWAY event and another event are scheduled to occur at the same time, the RELEASE.RUNWAY will occur first. In the LANDING and ARRIVAL.AT.IAF events, if two aircraft are due at the same time, then the one with the lowest remaining flying time is taken first.

Finally, in the preamble various variables are specified as statistics of interest for the simulation.

b. MAIN

First certain parameters are set. DAY is the day of the month when the simulation starts. Next, the permanent entities BASE and MSN.AREA are established and data pertaining to them is read. Finally, simulation control passes to the timing routine which executes the first scheduled event in the simulation.

c. TAKEOFF

In the event TAKEOFF, data pertaining to each aircraft is read in and processed; the aircraft enters the takeoff queue and takes off when runway conditions permit.

Takeoffs are scheduled externally at a local time for the particular base. Once the aircraft's data is read in the alpha-numeric values for its home base, destination base, alternate base and mission areas are converted to integer values. The current base of the plane becomes its

home base and mission times are converted to decimal days. Since takeoffs are scheduled at a local time and simulation time is taken to be GMT, the takeoffs are then rescheduled in six, seven or eight hours depending on the home base of the plane. This is a convenient happenstance for bases in the United States. If the model were applied to a world-wide or European situation this method of scheduling on local time and then rescheduling would not be practical.

When the rescheduled takeoff time occurs the plane is filed in the takeoff queue for the base. For the first takeoff a global variable TO.COUNT is equivalent to one and hence the printing of various statistics in 30 minutes is scheduled. The program considers four possible takeoff or landing conflict situations. That is, a conflict in this context means that a landing is scheduled in less than one minute ahead of the takeoff. In these situations the program chooses the courses of action as depicted in the following matrix:

Type of Landing	Takeoff Delay	Result
Touch and go	< 7 minutes	Touch and go has priority, takeoff rescheduled in 1 minute.
Touch and go	> 7 minutes	Takeoff occurs, landing aircraft makes a missed approach.
Full stop	< 7 minutes	Landing occurs, takeoff rescheduled in 1 minute
Full stop	> 7 minutes	Landing occurs, takeoff rescheduled at earliest opportunity.

The last check made prior to takeoff is to determine whether the runway is closed. If so the takeoff is canceled and data is printed out. Finally, the aircraft takes off and some processing is accomplished.

d. DEPARTURE.POINT

The departure point is one minute from the runway and is used to make decisions as to the next portion of the mission.

First, it is determined whether a plane is currently filed in a queue associated with the local pattern. This can occur if the plane arrives at the departure point after making a missed approach.

Next, if the plane is being diverted or is proceeding to another base or mission area, it goes enroute. Otherwise the plane remains in the local traffic pattern. Since heavier aircraft use the rectangular visual pattern and fighter-type aircraft use the overhead visual pattern, a test is made to determine the type of aircraft. For multi-engine aircraft, if the scheduled radar approaches have been completed and there is room in the rectangular pattern, the aircraft is filed in that pattern. A landing is then scheduled based on the number of aircraft in the pattern according to the rule TIME = number in queue + 4 minutes. If radar approaches are scheduled the aircraft enters the radar pattern. Time here is figured according to the rule TIME = (number in queue x 3) + 10 minutes. If the radar

pattern is at maximum the plane is diverted to the IAF to reenter the pattern. A flying time of 20 minutes was assumed. Similar rules were established for fighters with one exception. If the radar pattern is saturated the fighter is scheduled for a landing in nine minutes. This simulates leaving the pattern and reentering it.

e. ENROUTE

ENROUTE is used to route aircraft between bases, between mission areas and between bases and mission areas.

First, if the plane has been in a mission area it is removed from the refueling track, low-level route or delay area. If the plane is being diverted certain decision rules are called.

Next, if the plane is proceeding from base to base an ARRIVAL.AT.IAF is scheduled at the destination base and mission time and total flying time for the plane are updated to reflect the enroute flying time. In general, these times are caldulated in advance to reflect accurate totals at the next event for the aircraft. Two exceptions are approaches and takeoffs where these figures are updated after the next event.

In the next portion of the event if the aircraft has completed its mission area activities its final area exit time is set. Otherwise it is scheduled for the appropriate mission area. The mission option attribute was selected as a mechanism for determining to which mission area the plane

was to be routed. When the plane reached a mission area the mission option was set to zero. Further the scheduling rule adopted was as follows. If the plane was to be scheduled for one mission area, it became mission area #1. If the plane was to be scheduled for two mission areas, they became areas #1 and #2, and so forth.

f. MISSION

This event deals with the processing of the aircraft in three mission areas. Since the processing is similar for each area only one is discussed.

First, it is determined to which area (1, 2 or 3) the plane is arriving by checking the plane's mission option attribute. Once determined, program control passes to that area. If the plane is scheduled for a low-level route or a "delay" area it is filed in that set and scheduled to go enroute after its mission duration. Otherwise, the plane is filed in the refueling track. If it is not a tanker its tanker is sought out by looking for an aircraft in the refueling track set with a tail number which matches the plane's scheduled tanker tail number. Hence, an assumption here is that the tanker must be filed in the refueling track prior to the arrival of a receiver. If no match is found the receiver remains in the mission area for its mission duration. If a match is found the scheduled onload (in minutes) is added to the receiver's times and subtracted from the tanker's total flying time. An assumption here is

that an equivalent amount of "flying time" is transferred from one aircraft to the other.

Finally if all mission options for the plane are zero the mission stage is set to MSN.COMPLETE.

g. ARRIVAL.AT.IAF

In this event, a plane arrives at the IAF for its destination base and is scheduled for a landing. First, it is determined whether the plane is entering its home base pattern or is at another base. Next, the time of occurrence of the IAF is set. If the plane is scheduled for more than one IAF, the time for the last occurrence is set as the IAF time for the plane. Finally, the plane is filed in the IAF queue after it is scheduled for a landing according to the rule TIME = (number in queue x 6) + 15 minutes. This assumes that the IAF is located at a distance equivalent to 15 minutes of flying time from the base.

h. RELEASE.RUNWAY

The event returns the runway to an idle condition.

i. LANDING

In this event the plane gains control of the runway and makes a "touch and go" landing, a full stop landing or a missed approach.

First, the plane is removed from the queue it was last in and its times are updated.

Next, the landing event list is checked to see if there is another landing scheduled for the base within

one minute. If a landing "conflict" occurs then each landing in the landing list is delayed by one minute. To accomplish this, an attribute of the landing event called L.FLAG was used. When a plane's landing time is adjusted by one minute the L.FLAG is set to one. This indicates that no further adjustment is required at this time. Otherwise, the program would continually cycle through the LANDING list adding one minute to the scheduled landing time (TIME.A) for every plane.

If the runway is closed the plane makes a missed approach and is in a "divert" status.

Next, planes which are in transition training at a base other than their home base make a "touch and go" landing. Other planes are considered for a full stop landing based on their mission time remaining or their fulfillment of scheduled transition activities. It was assumed that mission time could be exceeded by as much as 30 minutes in order to fulfill transition requirements. If the full stop landing criteria are met the plane makes a full stop landing and various information concerning the plane is printed. Finally, the temporary entity aircraft is destroyed.

j. Routine CONVERSION

The routine converts the alpha-numeric values for bases and mission areas to integers. This is necessary since permanent entities are represented internally as integer subscripts.

k. Routine CURRENT.TIME

In this routine decimal days are converted to hours and minutes. First, current time is checked against the global variable DAY. If current time is one day later than DAY, DAY is increased by one. Next current time is changed to hours and minutes (GMT) and the home base of the aircraft is determined to give local time. If this local time is less than zero, 24 hours are added and the local day becomes one less than the Zulu day.

l. Routine DISTANCE

Here distance is calculated as the straight line distance between two points (x, y) and (x_1, y_1) according to the formula:

$$d = \sqrt{(x_1 - x)^2 + (y_1 - y)^2}$$

An assumption here is that the routing between two points (for example, from one base to another) is along a straight line connecting the two points.

m. Event HALF.HOUR.STATISTICS

This event is used to generate and print data relevant to each base's local traffic pattern every half hour. The first occurrence is triggered by the first takeoff in the simulation. Thereafter, the event is scheduled on a 30 minute basis until certain stopping criteria are met. Statistics generated here are the total number of aircraft

in a queue, the minimum number in the queue and the mean number in the queue for each queue at the base over the particular 30 minute period. To determine when to stop the totals for each queue at each base are checked. If all totals are zero a line is printed indicating that there has been no activity in the queues. Also a variable END.SIMULATION is increased by one. When a total of 48 half hour periods with no activity have occurred a stop simulation event is scheduled.

n. Event STOP.SIMULATION

Here final information and statistics are printed pertaining to each base. In addition to the half hour statistical data, statistics were accumulated over the entire period of the simulation. These were the maximum number of aircraft, the mean number of aircraft and the variance of aircraft in each queue of the local air traffic pattern. Other data printed here includes various base totals such as total number of takeoffs, landings, number of landing conflicts, and number of transient SAC aircraft.

III. INPUTS, OUTPUTS AND STRATEGY COMPARISON

A. INPUTS

The inputs to the model consisted of data pertaining to mission areas, bases, runway closures and individual aircraft. Advice in the formulation of input data for the simulation was acquired from crewmembers of the 924th Air Refueling Squadron, 328th Bombardment Squadron, 84th Fighter Interceptor Squadron, and instructors of the Strategic Air Command Central Flight Instructors Course at Castle AFB, California. In addition, interviews with Radar Approach Control (RAPCON) and tower personnel of the 2035th Communications Squadron at Castle AFB provided insights into the air traffic control situation at that base. Scheduling personnel at Castle AFB and March AFB, California were interviewed in ascertaining tactical unit procedures. Extensive coordination via telephone was accomplished with Fifteenth Air Force Directorate of Operations and Training (DOT) personnel in acquiring unit schedules and discussing particular situations for the model development. Eighth Air Force (DOT) personnel were also contacted for inputs concerning the Eighth Air Force bases. Inputs including mission areas, air refueling tracks, and mission profiles were purposely hypothesized but real world entities could easily be applied. In addition, routines dealing with specific strategies for the routing of aircraft upon base closures were considered "inputs."

Two such strategies are discussed later in this chapter. A typical set-up for a complete card deck is shown in Appendix A. Specific formats for the input are as follows. It should be noted that although column designators for specific items of data are listed that input was made via the SIMSCRIPT unformatted READ statement. Hence actual spacing of data on an input card is by user preference. For a description of specific items see the discussion of base and aircraft attributes in Chapter II.

1. Mission Areas

   Columns 4-6: x-coordinate (in minutes).

   Columns 8-10: y-coordinate (in minutes).

2. Bases

   Columns 2-4: 3-letter designator for the base.

   Columns 5-8: x-coordinate (in minutes).

   Columns 10-12: y-coordinate (in minutes).

   Columns 14-15: Total number of transient parking spaces available at the base.

   Column 21: Maintenance support (either B, K, O or X).

   Column 23: Maintenance support (either B, K, O or X).

   Columns 40-41: Maximum number of aircraft permitted in the radar queue.

   Columns 43-44: The maximum number of aircraft permitted in the rectangular queue.

   Columns 46-47: The maximum number of aircraft permitted in the overhead queue.

3. Aircraft

Each aircraft's input data was read in conjunction with the occurrence of the event TAKEOFF. There were two data cards associated with each aircraft.

Card 1:

Columns 2-8:	The word TAKEOFF.
Columns 13-14:	The day,
Columns 16-17:	hours and
Columns 19-20:	minutes (in local time) for the scheduled takeoff.
Column 22:	Aircraft type (1 for multi-engine, 2 for fighter-type).
Column 24:	Aircraft design (either B, K, F, T or O).
Columns 26-29:	Tail number.
Columns 31-33:	Home base (3-letter designator).
Columns 35-37:	Destination base (3-letter designator).
Columns 39-41:	Alternate base (3-letter designator).
Columns 43-46:	Mission time in decimal hours.
Column 48:	Mission profile or type (1-5).
Column 50:	Mission stage (0 [zero] for takeoff).
Columns 52-54:	Scheduled transition time (in minutes).
Columns 56-59:	Total flying time (in decimal hours).
Column 61:	Scheduled number of IAF approaches.
Column 63:	Scheduled number of missed approaches.
Column 65:	Scheduled number of radar approaches.

Column 68:	Scheduled number of visual approaches.
Column 71:	Scheduled number of "touch and go" landings.
Card 2:	
Columns 2-5:	Mission area #1 (4-character designator).
Columns 7-10:	Mission area #2 entrance time in hours and minutes (GMT).
Columns 12-15:	The total number of the paired aircraft for an air refueling.
Columns 17-19:	The onload/offload (in minutes) for an air refueling for area #1.
Column 21:	Mission option (1 for air refueling, 2 for low level, 3 for delay).
Columns 23-25:	The aircraft's duration (in minutes) for mission area #1.
Columns 27-30:	Mission area #2 (4-character designator).
Columns 32-35:	Mission area #2 entrance time in hours and minutes (GMT).
Columns 37-40:	The total number of the paired aircraft for an air refueling.
Columns 42-44:	The onload/offload (in minutes) for an air refueling for area #2.
Column 46:	Mission option (1, 2 or 3).
Columns 48-50:	The aircraft's duration (in minutes) for mission area #2.

Columns 52-55: Mission area #3 (4-character designator).

Columns 57-60: Mission area #3 entrance time in hours and minutes (GMT).

Columns 62-65: The tail number of the paired aircraft for an air refueling.

Columns 67-69: The onload/offload in minutes for an air refueling for area #3.

Column 71: Mission option (1, 2 or 3).

Columns 73-75: The aircraft's duration (in minutes) for mission area #3.

Column 78: * (asterisk). This is a SIMSCRIPT delineator to mark the end of the data read in with a particular event.

For card number 2, zeroes must be entered as default values if no activity is scheduled for the particular mission areas or if no air refueling is scheduled.

4. Change of Runway Status

Here, data is entered to prescribe the time and base where a runway closure will occur.

Columns 2-21: The characters CHANGE.RUNWAY.STATUS.

Columns 24-25: The day,

Columns 27-28: hours and

Columns 30-31: minutes (in GMT) for the runway closure.

Columns 50-52: The 3-letter designator of the base whose runway is being closed.

Column 72: * (asterisk). SIMSCRIPT delineator to mark the end of data.

B. OUTPUTS

The outputs of the program included information pertaining to each aircraft and statistical data relevant to each base. The data for an aircraft was printed with the occurrence of the final (full stop) landing for the aircraft. The output format was arranged on nine double lines as shown in Figure 2. The following information was included.

(1) Background Data: The aircraft's design, tail number, home base, current base and scheduled alternate base.

(2) Times: The aircraft's total flying time, mission time and transition time, all in minutes remaining.

(3) ATC Activities: The aircraft's scheduled transition activity versus activity actually accomplished.

(4) Mission Activities: This includes for each mission area the area designator, the aircraft's entrance and exit times (GMT) and for refueling activities the receiver (or tanker) design and tail number and the unload/offload in minutes.

(5) Event Times: The Aircraft's takeoff, IAF and final landing times in GMT and local time (with respect to the aircraft's home base).

Output information relevant to bases was made available through two events, HALF.HOUR.STATISTICS and STOP.SIMULATION.

```
DESIGN  TAIL.NUMBER  HOME.BASE CURRENT.BASE  ALTERNATE.BASE
  K        8454         FWH         FWH            DYS
TIMES (MINUTES REMAINING):                      ATC ACTIVITIES: IAFS VISUALS RADARS MISSED.APPROACHES TOUCH.AND.GOS
TOTAL.FLYING.TIME MISSION.TIME TRANSITION.TIME SCHEDULED        1       2       3            0                4
     513.95              -2.05          -38.05  ACCOMPLISHED     1       2       3            0                5
MISSION            ENTRY  EXIT                   RECEIVER/      TAIL      OFFLOAD/
ACTIVITIES   AREA  TIME   TIME  REFUELING:  TANKER DESIGN    NUMBER    ONLOAD(MINS)
    1        A7    2244   2339                                6583       120          EVENT TIMES: TAKEOFF IAF  LANDING
    2        0000     0      0                                   0         0               ZULU       1747    2    115
    3        0000     0      0                                   0         0               LOCAL      1147 1802   1915
```

Figure 2. Aircraft Data Output

1. Event HALF.HOUR.STATISTICS

In this event the total number, the maximum, the minimum and the mean numbers of aircraft in each of the queues (IAF, overhead, rectangular, radar and takeoff) at a base was presented each half hour. The first occurrence was scheduled with the first takeoff of an aircraft in the simulation. Thereafter the statistical values were reset to zero and another printing of values was scheduled for thirty minutes later. First, to reduce output printing the program checked to see if there had been any activity at any base during the half-hour interval. If there had been none, a single line stating "NONE" was printed and a counter, END.SIMULATION, was incremented. Information for all bases was printed if there had been activity at any base. The format was three double lines for each base with the name of the base and the total, maximum, minimum and mean number of aircraft for each queue at the base being printed as shown in Figure 3.

2. Event STOP.SIMULATION

This event was used to terminate the simulation and to print final data pertaining to each of the bases. As noted in the event HALF.HOUR.STATISTICS a counter termed END.SIMULATION was incremented where there was no activity at any base during a half-hour period. When this counter reached a value of 48 the event STOP.SIMULATION was triggered. Both statistical and general information was

HALF HOUR STATISTICS   TIME = 19.048611

BASE=SKA	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	1	0	0	3	1	TOTAL	2	0	0	9	2
MINIMUM	0	0	0	1	0	MEAN	.09	0.	0.	2.54	.23
BASE=BAB	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	0	0	0	TOTAL	0	0	0	0	0
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	0.	0.
BASE=MHR	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	0	0	0	TOTAL	0	0	0	0	0
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	0.	0.
BASE=SUU	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	0	0	0	TOTAL	0	0	0	0	0
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	0.	0.
BASE=MER	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	1	0	0	0	0	TOTAL	2	0	0	0	0
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	0.	0.
BASE=RIV	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	0	0	0	TOTAL	0	0	0	0	0
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	0.	0.
BASE=RCA	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	1	0	3	5	5	TOTAL	2	0	10	10	7
MINIMUM	0	0	0	3	3	MEAN	.03	0.	1.28	3.91	4.08
BASE=MIB	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	1	2	2	TOTAL	0	0	2	6	4
MINIMUM	0	0	0	0	0	MEAN	0.	0.	.17	1.22	1.1[illegible]
BASE=RDR	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	0	0	2	TOTAL	0	0	0	0	3
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	0.	1.01
BASE=OFF	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	0	0	0	TOTAL	0	0	0	0	0
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	0.	0.
BASE=IAB	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	1	0	0	0	1	TOTAL	2	0	0	0	1
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	0.	.23
BASE=LTS	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	1	1	1	TOTAL	0	0	5	4	2
MINIMUM	0	0	0	0	0	MEAN	0.	0.	.42	.93	.57
BASE=DYS	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	0	1	1	TOTAL	0	0	0	1	2
MINIMUM	0	0	0	0	0	MEAN	0.	0.	0.	.33	.50
BASE=FWH	TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF		TAKEOFF	OVERHEAD	RECTANGULAR	RADAR	IAF
MAXIMUM	0	0	1	1	0	TOTAL	0	0	2	3	0
MINIMUM	0	0	0	0	0	MEAN	0.	0.	.17	.75	0.

Figure 3. Half Hour Statistics

printed with regard to each base. Statistical data included the maximum, mean and variance associated with each queue at the base over the range of the simulation. Other information included total number of takeoffs, radar approaches, rectangular approaches, overhead approaches, IAF approaches, full stop landings, "touch and go" landings, missed approaches, takeoff conflicts, landing conflicts, available number of parking spots and number of transient aircraft for the base. A sample of the output for Altus Air Force Base is shown in Figure 4.

C. A SPECIFIC APPLICATION AS A DEMONSTRATION OF THE MODEL

1. Introduction

To demonstrate the potential uses of the model as a tool to solve regional scheduling problems the question of rescheduling aircraft upon base closures was investigated. In this setting a base's runway is closed and then a set of decision rules are applied to divert aircraft scheduled to arrive at that base. Here, two strategies were developed for the rescheduling of aircraft. Each strategy was implemented and the results were compared based on the flying time of the aircraft affected and the subsequent impact on the final destination bases of the aircraft with regard to the local traffic pattern, parking spaces and available maintenance support. Only SAC aircraft were considered in the two strategies . Normally SAC aircraft have a large fuel reserve which permits some latitude in decisions regarding diversions to an alternate base. Fighter aircraft operate

```
BASE = LTS
 QUEUES          MAXIMUM     MEAN       VARIANCE
 TAKEOFF            1          .00        .00
 RECTANGULAR        1          .00        .00
 OVERHEAD           1          .00        .00
 RADAR              4          .03        .05
 IAF                4          .02        .04
 TOTALS:   TAKEOFFS  RADARS  RECTANGULARS  OVERHEADS    IAFS    FSLS
               25.      48.        26.          10.       27.     27.
 TOTAL.T.G.S TOTAL.MISSED.APPS NO.TAKEOFF.CONFLICTS NO.LANDING.CONFLICTS
       84.          0.                   4.                  12.
 AVAILABLE.PARKING.SPOTS  NUMBER.OF.TRANSIENTS
          3                        2
```

Figure 4. Base Output Data

much closer to minimum fuel reserves. Thus, in this model fighters are assumed to land at the nearest available landing field. Since only SAC bases are represented in the model an additional base was created (Base #15). Fighters are thus diverted to this base when appropriate to emulate landing at the nearest available landing field. Under both strategies if an aircraft were performing transition training at a base other than its home base and its home base closes, it lands at the base where it is currently performing transition training. First, the two strategies are discussed; then, the specific scenario is outlined and, finally, the results are presented.

2. Strategy 1

a. Overview

In this strategy aircraft were rerouted to their scheduled alternate base if that base was open. If the alternate was also closed the aircraft were routed to the closest open base. To implement this strategy two general routines and three strategy specific routines were added to the model.

b. Routine POSITION

This routine gives the aircraft's current position based on its last known position and destination. The current position is based on a dead reckoning calculation with regard to the amount of time which has elapsed since the aircraft's last position was fixed.

c. Routine CLOSEST.BASE

In this routine the aircraft's position is given and the nearest open base is determined. First, the minimum distance to an open base is determined. Then the base which matches this minimum distance is found and becomes the output of the routine.

The following routines deal with the decision rules involved in Strategy 1. Three subsets of aircraft were considered: future takeoffs, aircraft in the local pattern and all other airborne aircraft.

d. Routine T.O.DECISION.RULES

First, the event TAKEOFF checks to see if the runway is closed. If it is closed the takeoff is canceled. Thus, these decision rules apply only to aircraft taking off from other bases which may later interact with a closed base. If the destination base for the aircraft is closed the alternate is checked. If it is open it becomes the destination for the aircraft. Otherwise, the routine CLOSEST.BASE is called and the nearest open base becomes the destination base.

e. Routine DEP.PT.DECISION.RULES

In the event LANDING if the base's runway is closed, the aircraft's EMER.DIVERT.FLAG is set to DIVERT. When the aircraft goes enroute this routine is called. Thus aircraft in the local pattern at the base remain in the pattern until reaching the base's departure point. This was done in this manner to emulate an orderly flow of aircraft through

the pattern prior to being diverted. If the aircraft is at its home base, first its alternate is checked and if the alternate is unavailable, the nearest base is selected. If the aircraft is at another base its home base is checked. If the home base is closed the aircraft becomes a mission type 3 (final landing at a base other than its home base) and its alternate is checked. Again, if the alternate is unavailable the nearest open base becomes the destination for the aircraft.

f. Routine AB.DECISION.RULES

This routine checks all other airborne aircraft. In this simulation each aircraft is always scheduled for a future event but may not be located in a particular set (refueling track, radar queue, etc.). Thus, the event lists are searched for the aircraft which are affected by the runway closure. In general the events which occur chronologically later in an aircraft's mission are checked first. This is of greater importance in Strategy 2 than here. Nevertheless, the general scheme is followed. Since aircraft in the landing pattern have been accounted for the first event checked is ARRIVAL.AT.IAF. First a series of tests is conducted to determine if the mission profile for the aircraft is affected. For example, if the aircraft is scheduled for transition training at another base and its home base is closed it becomes a mission type 3 (final landing at a base other than its home base). Next the aircraft's destination

base is checked. If that base is closed the aircraft's current position is determined. Next the alternate is checked and if necessary the nearest base is determined. This procedure is then continued for departures and landings at other bases (other than the one currently being closed). Finally, aircraft in mission areas or enroute to mission areas are checked.

3. Strategy 2

a. Overview

In this strategy aircraft were diverted to the nearest base which met the following two constraints: (1) that a parking space was available; (2) that the base could provide maintenance support for the aircraft.

To implement this strategy one general routine and three strategy-specific routines were included in the model.

b. Routine NRST.SUPPORT.BASE

This routine finds the nearest base which meets the two constraints discussed above. First, if the aircraft is a fighter it is routed to the nearest landing field (Base #15). Otherwise the closest base is first determined. If that base has remaining parking spaces and has matching maintenance support it becomes the destination base for the aircraft. If the base cannot meet the constraints its SEARCH.FLAG is set to one (so that it will not be considered further) and the next closest base is considered. This

process is repeated until a base which meets the constraints is found.

As in Strategy 1 the three strategy-specific routines here deal with future takeoffs, aircraft in the landing pattern of a closed base and other airborne aircraft. Future takeoffs are handled in a similar manner as in Strategy 1. A major difference in the strategies is that the order of consideration is of importance in Strategy 2. In Strategy 1 the choice of destination is made independent the choice for other aircraft. Here, once a suitable support base is located, the available number of parking spaces is decremented at that base. Consequently, aircraft in the landing pattern at the closed base are considered with other airborne aircraft in the routine AB.DECISION.RULES and the routine DEP.PT.DECISION.RULES is used merely to schedule the new ARRIVAL.AT.IAF.

c. Routine AB.DECISION.RULES

Here, the order of consideration of aircraft is important. Additionally, the order emulates the approximate relative fuel states of the aircraft involved. The order of consideration is aircraft in the landing pattern of a closed base, aircraft scheduled to arrive at an IAF, aircraft in the landing pattern of another base, aircraft enroute to a mission area, aircraft in a mission area and finally aircraft having just taken off. In each event list the processing is similar. If it is appropriate to consider the mission

profile this is done. Regardless, the nearest compatible support base is found and becomes the destination base of the aircraft. It should be noted that in this strategy the scheduled alternate base is not considered.

4. Scenario

The scenario developed here involved bases in the central valley of California. These are Castle AFB (MER), Mather AFB (MHR), Beale AFB (BAB) and Travis AFB (SUU). During certain periods of the year these bases are susceptible to being closed due to low-lying ground fog. In the first instance Castle was simulated to be closed at 1900Z (1100 local time). In the next instance Castle was closed at 1900Z followed by Mather being closed at 1930Z. This pattern was then continued with Beale and finally Travis being closed at half hour intervals. For inputs, a total of 269 takeoffs were scheduled at the 14 bases ranging in time from 1310Z to 0406Z of the next day. Castle was scheduled for 19 takeoffs, Mather for 24, Beale for 12 and Travis for 14.

5. Results

a. Case 1, Castle Closes at 1900Z

In this case a total of seven aircraft were diverted. Under Strategy 1, six aircraft were diverted to March AFB, California. The seventh aircraft was scheduled for transition training at Fairchild AFB, Washington and hence landed there. The average total airborne time for

the seven aircraft was 438 minutes. The available number of parking spaces was exceeded by one at March.

Under Strategy 2, four aircraft were diverted to Beale, one to Mather, one to Travis and one to Fairchild. The average airborne time for the seven aircraft was 418 minutes.

b. Case 2, Castle Closes at 1900Z, Mather Closes at 1930Z

In this instance a total of 20 aircraft were diverted, seven from Castle and thirteen from Mather. With Strategy 1 three were diverted to Beale, ten to Travis, six to March and one to Fairchild. The average total airborne time per aircraft was 385 minutes. In two cases, the number of available parking spaces was exceeded. At March, with five available spots six aircraft landed. At Travis, with one available parking spot two aircraft landed.

Under Strategy 2 eight aircraft were diverted to Beale, four to Travis, four to March, three to Fairchild and one to Dyess AFB, Texas. The average total airborne time per aircraft was 370 minutes. The number of available parking spots was exceeded at Beale by three aircraft and at Travis by two aircraft.

c. Case 3, Castle Closes at 1900Z, Mather Closes at 1930Z, Beale Closes at 2000Z

In this case a total of 22 aircraft were diverted. Under Strategy 1 six aircraft were diverted to March, one to Fairchild and 15 to Travis. The average airborne time per aircraft was 392 minutes. Parking spaces were exceeded at

two bases. At March they were exceeded by one and at Travis by 14 aircraft.

With Strategy 2 five aircraft were diverted to Travis, five to March, four to Ellsworth AFB, South Dakota, six to Fairchild and two to Dyess. The average airborne time per aircraft was 403 minutes. Available parking spaces were exceeded at two bases. At Fairchild they were exceeded by one and at Travis by four aircraft.

d. Case 4, Castle Closes at 1900Z, Mather Closes at 1930Z, Beale Closes at 2000Z, Travis Closes at 2030Z

In this last case a total of 26 aircraft were diverted, seven from Castle, thirteen from Mather, two from Beale and four from Travis.

Under Strategy 1 twenty aircraft were diverted to March, four to Travis (prior to its closing) and two to Fairchild. The average airborne time was 411 minutes. Parking spaces were exceeded at March by 14 aircraft. Although four aircraft were diverted to Travis, the four Travis aircraft were diverted to other bases.

Under Strategy 2, two aircraft were diverted to Travis, five to March, six to Fairchild, five to Ellsworth, four to Dyess, two to Altus AFB, Oklahoma and two to Minot AFB, North Dakota. The average airborne time was 428 minutes. Available parking spaces were not exceeded at any base.

A synopsis of the four cases is as shown in Table II.

TRANSIENT AIRCRAFT

	Base: (Number)	RIV (5)	SKA (5)	SUU (2)	BAB (5)	DYS (5)	RCA (5)	LTS (2)	MIB (5)	Average Flying Time (Mins)
Case 1	Strategy 1	6	1	0	0	0	0	0	0	438
	Strategy 2	0	1	1	4	0	0	0	0	418
Case 2	Strategy 1	6	1	10	3	0	0	0	0	385
	Strategy 2	4	3	4	8	1	0	0	0	370
Case 3	Strategy 1	6	1	15	0	0	0	0	0	392
	Strategy 2	5	6	5	0	2	4	0	0	408
Case 4	Strategy 1	20	2	4	0	0	0	0	0	410
	Strategy 2	5	6	2	0	4	5	2	2	428

Table II. Synopsis of Results

6. <u>Conclusions</u>

In comparing the two strategies the impact of the final landing bases and aircraft airborne times are discussed.

a. Impact on Bases

Under both strategies continuity of maintenance support was maintained. That is, there were no discrepancies between aircraft type and maintenance support available at the final landing base. This is due in part to the fact that of the eight bases affected in the four cases six had maintenance support capabilities for both B-52s and KC-135s.

The most severe impact on bases was evidenced in Case 3 for Travis and Case 4 for March. Under Strategy 1 in Case 3 a total of 15 transient aircraft landed at Travis. The set of graphs depicted in Figures 5 and 6 show maximum numbers of aircraft in the radar and IAF queues at Travis at half-hour intervals starting at 1940Z under Strategy 1 and Strategy 2. Another indicator was the number of landing conflicts at Travis. Under Strategy 1, eighteen landing conflicts while under Strategy 2, nine landing conflicts occurred. In addition to the increased traffic load in the local pattern there is an implication of increased demand on base resources with respect to providing maintenance support and security for the 15 transient aircraft.

Similar results occurred with March AFB under Strategy 1 in Case 4. Again the sets of graphs depicted in Figures 7 and 8 indicate the maximum number of aircraft

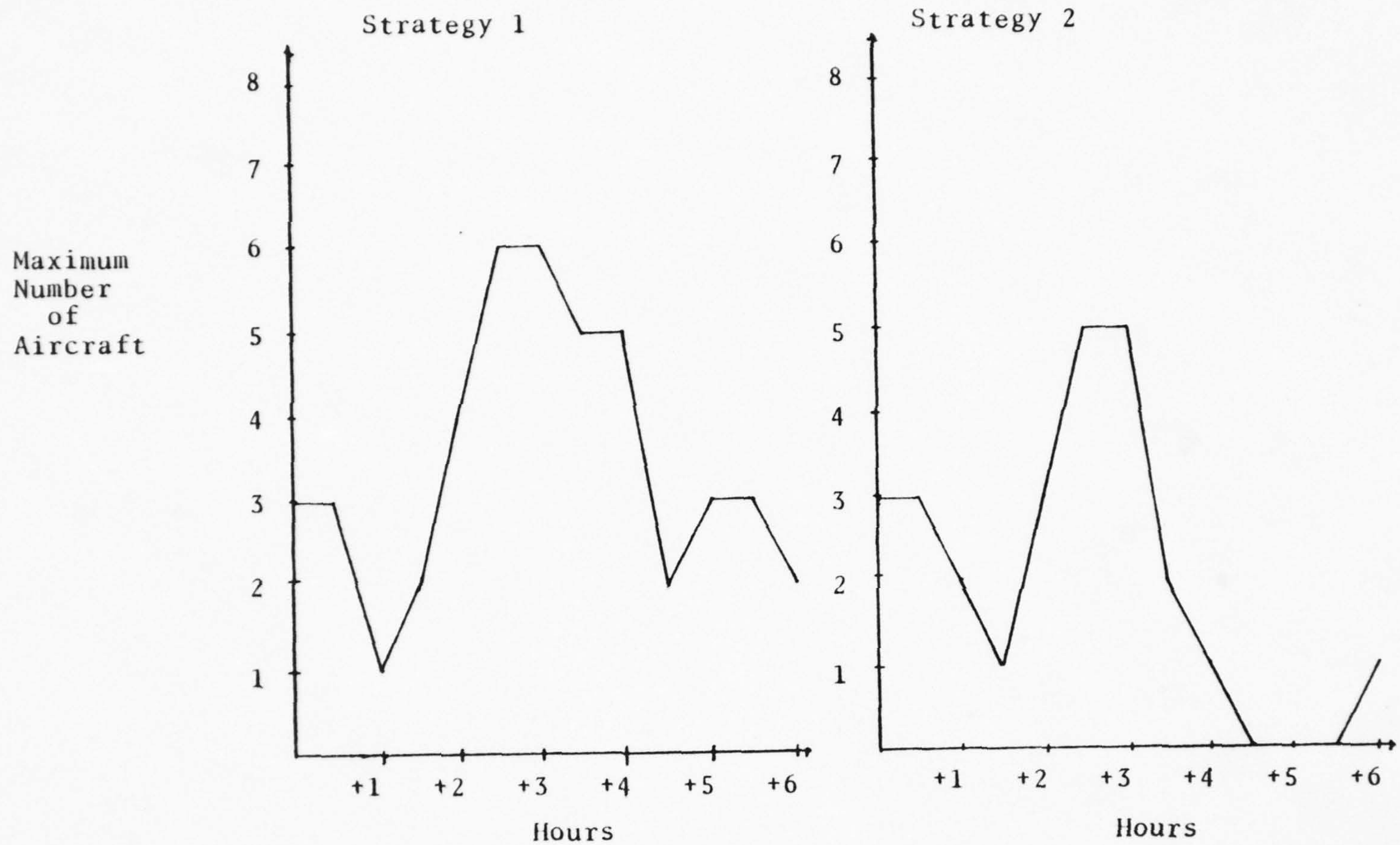


Figure 5. Radar Queue at Travis

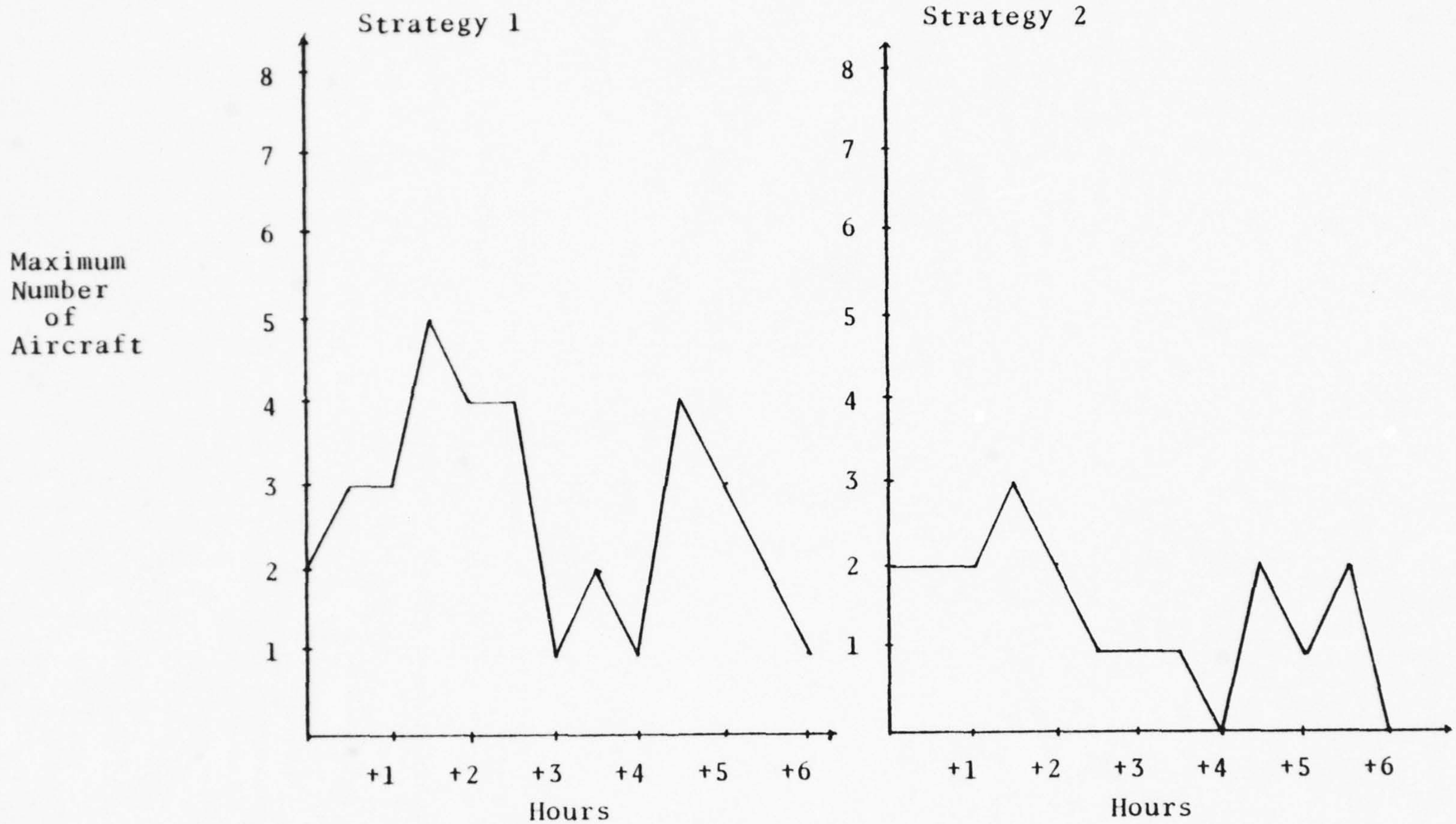


Figure 6. IAF Queue at Travis

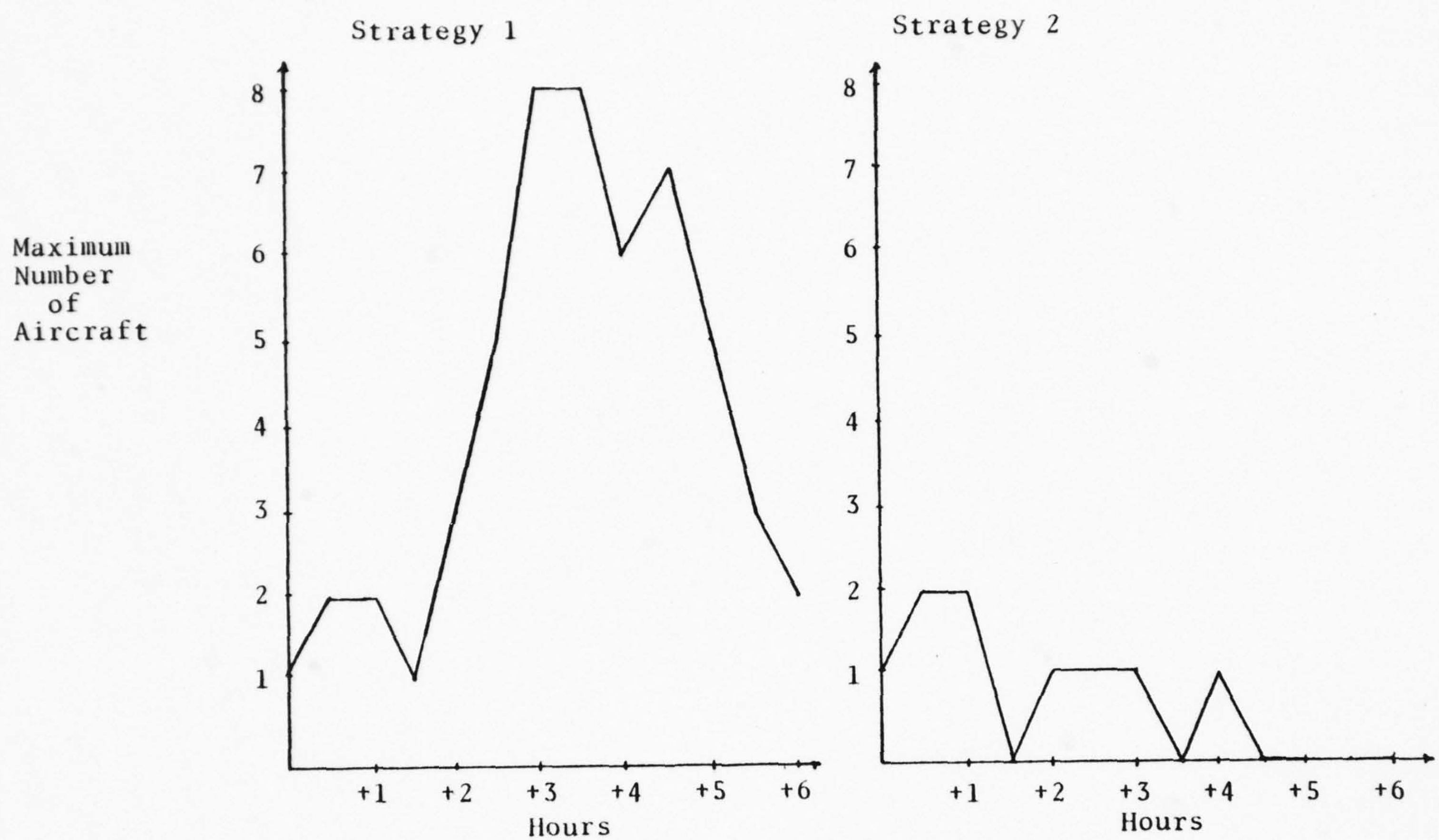


Figure 7. Radar Queue at March

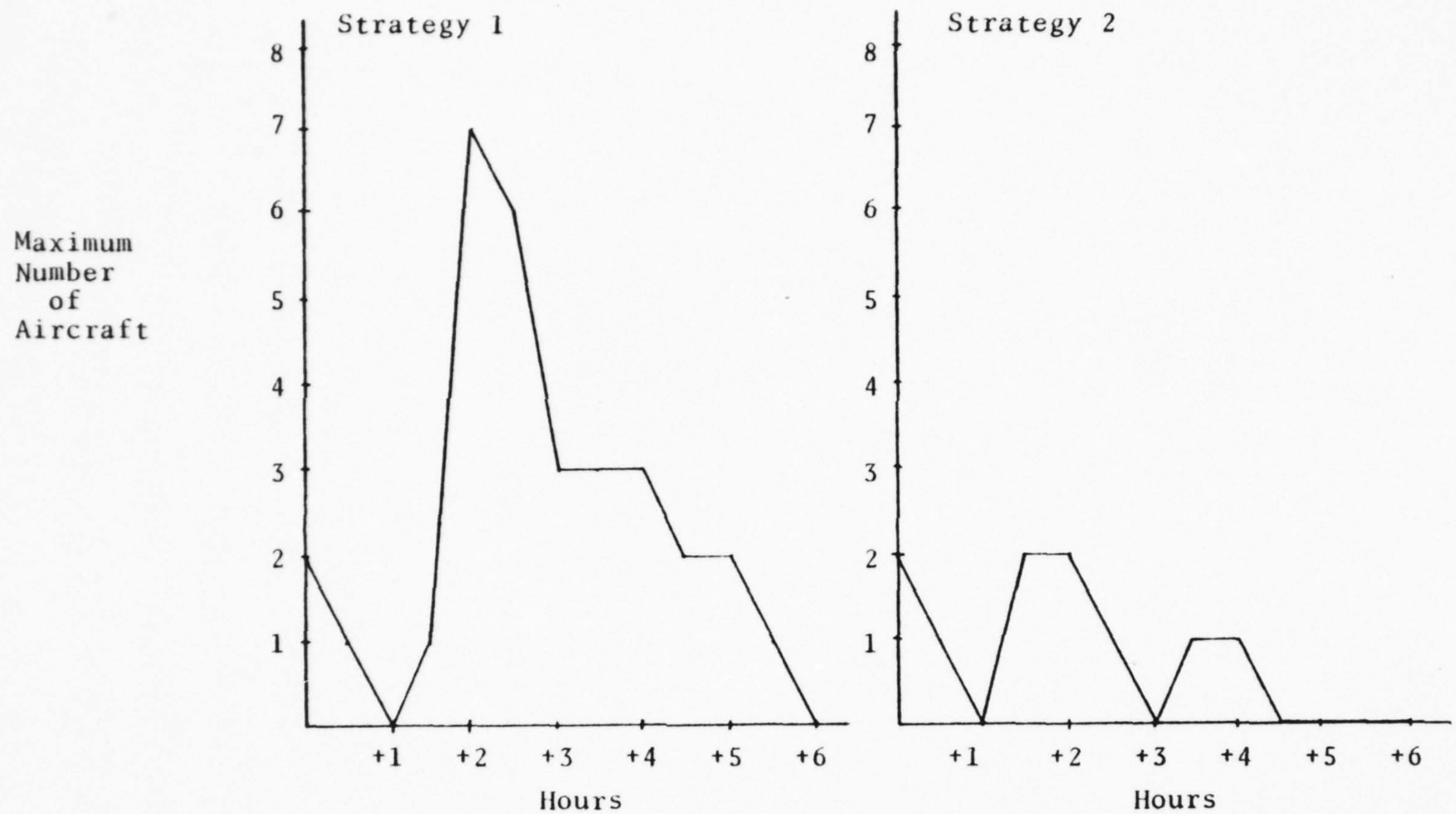


Figure 8. IAF Queue at March

in the radar and IAF queues at March at half hour intervals starting at 1940Z. Another comparison is the number of landing conflicts which occurred at March. Under Strategy 1 there were 20 landing conflicts while under Strategy 2 there were two landing conflicts. Again there is a strong implication of an increased demand on base resources due to the influx of a large number of transient aircraft.

b. Average Airborne Times per Aircraft

Under Case 1 and Case 2, Strategy 2 fared better, by 20 minutes per aircraft in Case 1 and 15 minutes in Case 2. These results were due in part to the selection of an alternate base by the scheduler. In these two cases most aircraft under Strategy 1 were diverted to their scheduled alternate. Hence, the scheduler can make an impact on the model's outcome by selecting a more diverse set of alternate bases.

Under Case 3 and Case 4, Strategy 1 provided lower average times, by 16 minutes with Case 3 and 18 minutes with Case 4. Although in general the set of bases selected under Strategy 1 was geographically chosen with regard to flying time the aircraft experienced increased delays in the local traffic pattern. For example, at March under Case 3, the seventh aircraft in the IAF queue would experience a delay of 51 minutes between the IAF point and a subsequent landing. Similarly the eighth aircraft in the radar queue would experience a time of 34 minutes to complete the pattern.

In conclusion then, Strategy 1 appears to offer little compensation for the adverse impact it imposes on

certain diversion bases. Although the scheduler can improve results through a more diverse selection of alternates, Strategy 2 seems to provide a more logical approach through consideration of available base resources with only a small overall difference in average airborne time.

IV. LIMITATIONS, EXTENSIONS, AND RECOMMENDATIONS

A. INTRODUCTION

The computer simulation model was designed to evaluate two decision strategies for air traffic management at a regional level. The simplifying assumptions and deterministic nature of the model were somewhat unrealistic of true air traffic operations but preserved the essence of the regional rescheduling problem. This chapter discusses limitations of the model and proposes extensions and recommendations.

B. LIMITATIONS

The greatest limitation of the model was its deterministic nature. For example, the minimum of one minute between takeoffs and landings at a base represented an average rather than actuality. In reality, these times vary according to aircraft sizes, weights, speeds, and other factors. Often formation takeoffs or landings are accomplished wherein two or more aircraft simultaneously utilize the runway. Also, heavy or "jumbo" jet rules apply at certain times wherein minimum intervals between aircraft are extended because of wake or air turbulence effects.

A second unrealistic characteristic of the model was the movements of aircraft at average speeds over certain distances instead of computational rates based on aircraft gross weight, configuration, altitude, and other factors. For cruise

flight, the assumption was an average speed of four hundred knots true air speed (KTAS) and a no-wind condition. The aircraft movements from the initial approach fix and in radar and visual patterns were computed by simple calculations involving numbers of aircraft times arbitrary constants plus other arbitrary constants. The maximum numbers of aircraft allowed in each queue represented traffic saturation for the operating conditions. In reality, the cruise and transition (or traffic pattern) speeds vary widely depending on aircraft type, altitude, configuration, gross weight, and other factors. However, time spacing required for air traffic control compensates for these phenomena. For example, a typical descent for fighter aircraft from the initial approach fix is at 280 knots indicated air speed (KIAS) with only the speed brake extended (the other drag devices including the landing gear and flaps are retracted). When the pilot changes the aircraft configuration for landing by lowering the landing gear and flaps, the airspeed changes drastically. The final approach speed may be about 120 to 170 knots for typical approaches. However, the air traffic controllers and other pilots maneuver the aircraft so that minimum time criteria are satisfied and the orderly flow of traffic is maintained. So in practice a faster aircraft following a slower aircraft flies an outside or larger pattern and therefore a longer distance over the same time interval.

A third unrealistic characteristic was the characterization of a mission area as a point. While this served the purposes of the simulation, time elapses at a point for an airborne aircraft cannot be realistic for the missions considered. However, the low-level Olive Branch routes can be traversed in only one direction so an attempt was made to measure distance and time from the appropriate ends of the routes. Also, approximate midpoints for the hypothetical air refueling tracks were assumed so that refuelings in either direction were possible.

A fourth and final unrealistic characteristic of the model concerned the closures of the runways. In reality, most closures occur with some forewarning and some degradation of capabilities. Also, most closures are necessarily temporary but vary widely in the amount of time the runway is closed. Often aircraft are ordered into holding patterns in anticipation of the reopening of the runway.

C. EXTENSIONS

Some modifications of the model are required in order to utilize it in investigation of many real-life problems. The probabilistic enhancements necessary to emulate actual air traffic operations involve analyses of specific operations. For instance, each base has air traffic characteristics peculiar to its operations. These are functions of geography, weather, safety and environmental considerations, types and missions of aircraft, and other factors. The actual bases

utilized in the simulation could be characterized by base and aircraft movement functions or transforms. These transforms could be placed in look-up tables and called from the program based on the state of the simulation. Likewise, the operating parameters of the aircraft such as fuel capacities, consumption rates, speeds, and altitudes can be characterized as transforms and accessed from look-up tables as required.

Besides the analyses to determine the probabilistic natures of the bases and aircraft, it is necessary to determine the probabilistic distributions for scheduling determinations. Actual takeoff times, arrivals at certain points, and landing times are probably described by exponential distributions but would have to be determined. Aircraft emergency and abort occurrences should be integrated into the model also. Their probability distributions are likely to be uniform but would have to be determined.

With these probabilistic enhancements and other individualized modifications, the utility of the model as a departure point for other theses is great. Indeed, the development of the enhancements could constitute a thesis investigation. However, logical applications of the model include CONUS and world-wide SAC air traffic management or command and control (C^2) as well as other strategic and tactical investigations.

1. Strategic

a. MAC Worldwide Logistics

An appropriate strategic application would be the worldwide logistics airlife problem of the Military

Airlift Command (MAC). The attribute-structure of the SIMSCRIPT language allows the characterization of all aspects of the size and weight-carrying capabilities of the cargo aircraft. The base structure of the basic model is probably appropriate for most applications, especially for the C-5A and C-141A aircraft which are MAC's strategic airlifters. The base location and cruising airspeed factors would have to be changed but the application should be straightforward after these changes.

If the decision to modify the existing fleet of 271 C-140A's with an air refueling capability is made, the air refueling feature of the model would take on added importance. Currently, the air refueling capability of the 77 C-5A "Galaxy" makes it the only truly "global" airlift aircraft in the U.S. inventory.

Such investigations as how to move CONUS ground and air units to Europe and other places could be made. Additionally, the numbers and mix of airlift and tanker aircraft required for the airlift could be determined. Also a time-ordered analysis would indicate where materiel should be pre-positioned for various options or contingencies. Traditional operations research (OR) techniques such as linear programming (LP) and network analysis would be applicable to these analyses.

b. *Tanker Task Forces*

A second strategic application has already been mentioned. It is tanker support to airlift operations but

could easily be extended to tanker task forces and movements of fighter aircraft over long distances. Likewise, it could be applied to strategic bomber operations. Again, as with the airlift problem, optimum locations for tankers and fuels could be determined for various options or contingencies.

The feature of the model which makes it amenable to the above applications is its modular structure. The elements of the simulation can be developed in sets or modules so that substitution or the development of a new module can be easily incorporated. For example, the bases can be described in a general nature as in the basic model or developed individually by transforms and look-ups. Likewise, the aircraft can be described by their individual attributes and by transforms and look-ups. Also, the events and routines can be added or deleted as required to describe the applicable air traffic management situation. For example, the Military Airlift Command may operate out of hundreds of airfields throughout the world but may be interested only in the bases to be utilized by the C-5A aircraft during one week's flying activities. All bases in the world would be on file in a database and accessed by attributes such as name, location, length of runway, governmental status, or any other trait. A front-end processor or routine would determine the applicable bases and aircraft. The appropriate events and routines as well as the scheduling inputs would be developed. From this point, the simulation of the C-5A weekly flying schedule would be straightforward.

2. Tactical

a. Tactical Logistics

With this description of the enhanced model, it becomes obvious that tactical theater logistics investigations follow directly from the strategic ones. The 270-plus MAC C-130 aircraft qualify as tactical airlifters and the "Hercules" can be described by attributes and transforms. Because several tactical airlift missions can be flown in one day, some input modifications are necessary. But these are easily applied in modular form. Also, the C-130 can operate out of more airfields than their strategic counterparts so the bases must be more fully described. For example, the C-130 can operate on dirt strips. However, the capability of attribute development in the SIMSCRIPT language easily allows for this distinction.

As with the strategic case, logistics placements can be determined for optimum airlift operations. In fact, the interface between strategic and tactical airlift operations can be addressed so that the entire airlift operation can be optimized.

b. Fighter Operations

In addition to tactical airlift, tactical fighter and reconnaissance operations can be accommodated by the model. Again the aircraft and applicable airfields would have to be described in attributes and transforms. In this instance, the point-location aspect of the basic model's air refueling areas may be appropriate. Normally in hostile

theater operations, tankers orbit on the periphery of the hostile area at locations termed "anchors." Tactical aircraft are vectored to the tankers or are scheduled for pre- or post-strike refueling at appropriate anchors.

Obviously, theater tactical warfare is much more complex than CONUS training or airlift operations because of uncertainties, the dynamic nature of the tactical situation, and marginal operating parameters of tactical aircraft. However, the essence of the tactical situation could be abstracted for certain applications and the model could provide insights into operational or planning factors. For example, the tanker support required for theater operations could be determined for various levels of activity. Or analyses concerning activity levels necessary to launch, recover and reconfigure tactical aircraft could be made. Certain critical studies such as plans for launch and recovery of aircraft while the airfield is under enemy attack could be accomplished. Results of these investigations would be alert or readiness determinations, traffic spacing criteria, air traffic control capacities, aircraft diversion requirements, and many others. Much imaginative thinking would be required in order to derive survival and other probability distributions but approximations would probably provide insights into tactical air management problems.

c. Carrier Operations

Another investigation into tactical air management is U.S. Navy aircraft carrier operations. While major

changes to the model may be necessary, certain applications such as diversions from a damaged carrier might be appropriate. Also, air refueling and defensive operations appear to be good candidates for investigations. Of course, the mobile aspect of carrier operations would have to be accounted for in most applications.

D. RECOMMENDATIONS

Having discussed specific strategic and tactical applications of the model, it is appropriate to make recommendations for utilization of the model. First, the specific problem to be addressed must be determined. Then the essential operating variables and relationships should be determined. Dependencies and probabilities of occurrences should be transformed into functions and distributions. Modifications to the model, normally in modular form, should then be made so that the essence of the air traffic management situation is fully described. An exception to the modular change occurs in the revised scheduling statements but SIMSCRIPT allows straightforward scheduling statements. For example, the statement to schedule an aircraft to the initial approach fix of a base may be the following:

```
SCHEDULE AN ARRIVAL.AT.IAF AT
TIME.V + EXPONENTIAL.F (MEAN,1).
```

The arrival would be at current simulation time (TIME.V) plus an amount of time determined by a library function (EXPONENTIAL.F) which draws a real number from an exponential

distribution with a mean of MEAN, a real variable, and a sequence of numbers determined by an initialization or "seed" from a stream determined by the integer 1.

After appropriate modifications are made to the model, the inputs and element look-up tables or matrices should be developed. Then the output requirements are determined and appropriate formatted or unformatted PRINT statements written. The inevitable debugging follows and then fully developed simulation. Appropriate statistics and sensitivity analyses probably would require changes in the model or input and result in an iterative process.

Another recommendation for the utilization of the model would be the development of interactive capability including graphics and online query. Inherent in this interactive capability is the requirement for an appropriate database and the means to access it. While the data and structure of the database could constitute another major investigation, suffice it to say that the speed, efficiency, reliability, accuracy and information requirement of the interaction would be highly dependent on the application. For example, if the information were used for contingency planning or tactics development for several years into the future, the speed and efficiency would not necessarily be critical. However, in a dynamic tactical situation such as correlation of aircraft tracks in the operation of the U.S. Air Force E-3A "Sentry" airborne warning and control system (AWACS) aircraft, the speed and efficiency are crucial. In this case, graphics

with efficient query capabilities such as penlights, tracking handles, function keys, or "joysticks" would be appropriate. The real-time query capability necessary for the track-correlation of the E-3A has been demonstrated in such computer network projects as the DOD Advanced Research Projects Agency (ARPA) network. The time-phased tracks necessary for the correlation could be provided by simulation with an enhanced model. Reports by pilots or other crewmembers could update positions or intentions. The inherent capability of the AWACS or the ground-based Combat Reporting Center (CRC) portion of the Tactical Air Control System (TACS) could update known friendlies' positions. In addition, the advanced capabilities promised in the Joint Tactical Information Distribution System (JTIDS) including relative navigation and programmed reporting could provide the same information with no human intervention.

The ultimate capability of the model would reside in the development of decision strategies by artificial intelligence (AI). The computer would develop this capability through emulation of the commander's decision processes while using the model as an operational decision aid. At some point, the computer's ability at developing strategies could be validated. The development of the decision strategies would be dependent on the status of the bases and the aircraft. The high-speed report capability of JTIDS could constantly update the database so that the computer could continually

determine the appropriate decision strategy. Appropriate command and control means could implement the decision strategy in near real-time.

E. SUMMARY

The thesis evolved from an investigation into the feasibility of developing integrated base and regional scheduling methods. Because of inherent complexities, a subset of the regional problem was selected as the subject of the thesis. The rescheduling of aircraft to alternate bases when runway closures occurred was investigated. A digital computer simulation model for air traffic management was developed and was described in Chapter II. The inputs and outputs of the simulation and the evaluation of two decision strategies were presented in Chapter III.

Limitations of the model were presented in Chapter IV. Applicable guidelines to extend the model to several strategic and tactical applications follows. Finally, some recommendations for utilization of the model were presented. These included methodology for implementation of the model and a discussion of interactive possibilities. Recent technological advances were mentioned as well as a possible ultimate outcome for the model.

APPENDIX A

PROGRAM DECK FORMAT

The following listing indicates the proper deck set-up for the model.

JOB CARD (with TIME = 5)

//EXEC SIM25CLG, REGION = 250K (If the number of aircraft in the simulation exceeds approximately 100, the region should be set to 350K.)

```
//SIM.SYSIN DD *
{PROGRAM DECK
{DECISION MODULE ROUTINES
/*
//GO.SYSIN DD *
{BASE DATA
{MISSION AREA DATA
/*
//GO.SIMU07 DD *
{AIRCRAFT DATA
/*
//GO.SIMU08 00 *
{RUNWAY CLOSURES DATA
/*
ORANGE CARD
```

APPENDIX B

LIST OF ABBREVIATIONS

DOD - Department of Defense

HQ - Headquarters

USAF - United States Air Force

ATC - Air Traffic Control

FAR - Federal Aviation Regulation

PPR - Prior Permission Required

IFR - Instrument Flight Rules

FLIP - Flight Information Publications

SAC - Strategic Air Command

C^2 - Command and Control

CONUS - Continental United States

NAF - Numbered Air Force

AFB - Air Force Base

ICAO - International Civil Aviation Organization

RAPCON - Radar Approach Control

DOT - Directorate of Operations and Training

KTAS - Knots True Airspeed

KIAS - Knots Indicated Airspeed

OB - Olive Branch Training Route

MAC - Military Airlift Command

OR - Operations Research

LP - Linear Programming

AWACS - Airborne Warning and Control System

ARPA - Advanced Research Projects Agency

CRC - Combat Reporting Center

TACS - Tactical Air Control System

JTIDS - Joint Tactical Information Distribution System

AI - Artificial Intelligence

```
PREAMBLE
 NORMALLY, MODE IS REAL
 DEFINE IDLE TO MEAN 0
 DEFINE BUSY TO MEAN 1
 DEFINE CLOSED TO MEAN 2
 DEFINE MULTI.ENGINE TO MEAN 1
 DEFINE FTR.TYPE TO MEAN 2
 DEFINE DIVERT TO MEAN 1
 DEFINE EMERGENCY TO MEAN 2
 DEFINE JUST.TOOKOFF TO MEAN 0
 DEFINE MSN.COMPLETE TO MEAN 4
 DEFINE HOMEBASE.PATTERN TO MEAN 5
 DEFINE OTHER.BASE.PATTERN TO MEAN 6
 DEFINE BOMBER TO MEAN B
 DEFINE FIGHTER TO MEAN F
 DEFINE TANKER TO MEAN K
 DEFINE TRAINER TO MEAN T
PERMANENT ENTITIES
 EVERY MSSN.AREA HAS
  A X.COOR,
  A Y.COOR AND MAY OWN
  A REFUELING.TRACK,
  A DELAY.AREA,
  A LOW.LEVEL.ROUTE
 EVERY BASE HAS
  A NAME,
  A X.POS,
  A Y.POS,
  SOME TRANSIENT.PARKING.SPACES,
  A TTL.NO.TRANSIENTS,
  AN AVAIL.NO.TPS,
  A SEARCH.FLAG,
  A 1.MAINT.SUP.CAP,
  A 2.MAINT.SUP.CAP,
  A 3.MAINT.SUP.CAP,
  A RUNWAY,
  A MAXIRADAR.QUEUE,
  A MAXIRECTANGULAR.QUEUE,
  A MAXIOVERHEAD.QUEUE,
  A TTL.RECTANGULAR,
  A TTL.RADAR,
  A TTL.OVERHEAD,
  A TTL.TAKEOFFS,
  A TTL.LANDINGS,
  A TTL.IAF,
  A TTL.T.G.LANDINGS,
  A TTL.FSL.LANDINGS,
  A TTL.MISSED.APPROACHES,
```

```
 A NO.TAKEOFF.CONFLICTS,
 A NO.LANDING.CONFLICTS
 AND OWNS
 THE IAF.QUEUE,
 THE TAKEOFF.QUEUE,
   THE RECTANGULAR.QUEUE,
   THE OVERHEAD.QUEUE,
   THE RADAR.QUEUE
TEMPORARY ENTITIES
 EVERY AIRCRAFT HAS
   A TYPE,
   A DESIGN,
   A TAIL.NUMBER,
   A LOCAL.TIME,
   AN A.HOME.BASE,
   A HOME.BASE,
   AN A.CURRENT.BASE,
   A CRNT.BASE,
   AN A.DESTINATION.BASE,
   A DESTINATION.BASE,
   AN A.ALTERNATE.BASE,
   AN ALTERNATE.BASE,
   A MISSION.TIME.REMAINING,
   AN EMER.DIVERT.FLAG,
   A LKT,
   A X.LAST.POS,
   A Y.LAST.POS,
   A TRANSITION.TIME.REMAINING,
   A TOTAL.FUEL.REMAINING.TIME,
   SOME IAF.APPROACHES,
   SOME MISSED.APPROACHES,
   SOME RADAR.APPROACHES,
   SOME VISUAL.APPROACHES,
   SOME T.G.LANDINGS,
   SOME SCH.IAF.APPROACHES,
   SOME SCH.MISSED.APPROACHES,
   SOME SCH.RADAR.APPROACHES,
   SOME SCH.VISUAL.APPROACHES,
   SOME SCH.T.G.LANDINGS,
   SOME A.IAFS,
   SOME A.MISSED,
   SOME A.RADARS,
   SOME A.VISUALS,
   SOME A.T.GS,
 AN A.1.MSN.AREA,
 AN A.2.MSN.AREA,
 AN A.3.MSN.AREA,
 A MSN.1.AREA,
```

```
A MSN.2.AREA,
A MSN.3.AREA,
A M.1.AREA.TIME,
A M.2.AREA.TIME,
A M.3.AREA.TIME,
A 1.RCVR.TNKR.TAIL.NO,
A 2.RCVR.TNKR.TAIL.NO,
A 3.RCVR.TNKR.TAIL.NO,
A 1.ONLOAD.OFFLOAD,
A 2.ONLOAD.OFFLOAD,
A 3.ONLOAD.OFFLOAD,
A 1.MSN.OPTION,
A 2.MSN.OPTION,
A 3.MSN.OPTION,
A 1.DURATION,
A 2.DURATION,
A 3.DURATION,
A LAST.MSN.AREA,
  A Z.TAKEOFF,
  A Z.IAF,
  A Z.LANDING,
  A L.TO,
  A L.IAF,
  A L.LNDG,
  A GMT,
  A 1.ENTRY,
  A 2.ENTRY,
  A 3.ENTRY,
  A 1.EXIT,
  A 2.EXIT,
  A 3.EXIT,
  A 1.R.T.DESIGN,
  A 2.R.T.DESIGN,
  A 3.R.T.DESIGN,
  A MSN.STAGE,
  A MSN.TYPE,
  A TIM1.IN.TAKQUEUE,
  A TIM2.IN.OVRQUEUE,
  A TIM3.IN.RECQUEUE,
  A TIM4.IN.RADQUEUE,
  A TIM5.IN.IAFQUEUE
  AND MAY BELONG TO
  A TAKEOFF.QUEUE,
  AN IAF.QUEUE,
  AN OVERHEAD.QUEUE,
  A RECTANGULAR.QUEUE,
  A RADAR.QUEUE,
  A DELAY.AREA,
```

```
   A LOW.LEVEL.ROUTE,
   A REFUELING.TRACK
EXTERNAL EVENTS ARE TAKEOFF
   AND CHANGE.RUNWAY.STATUS
EXTERNAL EVENT UNITS ARE 07 AND 08
EVENT NOTICES
   INCLUDE
   HALF.HOUR.STATISTICS AND
   STOP.SIMULATION
  EVERY RELEASE.RUNWAY HAS AN AIRFIELD
   EVERY TAKEOFF HAS A PLANE1
   EVERY DEPARTURE.POINT HAS A PLANE2
   EVERY ARRIVAL.AT.IAF HAS A PLANE3 AND A FUEL1
   EVERY LANDING HAS A PLANE4, A FUEL2 AND A L.FLAG
   EVERY .MISSION HAS A PLANE5
   EVERY ENROUTE HAS A PLANE6
DEFINE DISTANCE AS A REAL FUNCTION
DEFINE END.SIMULATION AS AN INTEGER VARIABLE
DEFINE DAY AS AN INTEGER VARIABLE
DEFINE TO.COUNT AS AN INTEGER VARIABLE
DEFINE PLANE1 AS AN INTEGER VARIABLE
DEFINE PLANE2 AS AN INTEGER VARIABLE
DEFINE PLANE3 AS AN INTEGER VARIABLE
DEFINE PLANE4 AS AN INTEGER VARIABLE
DEFINE PLANE5 AS AN INTEGER VARIABLE
DEFINE PLANE6 AS AN INTEGER VARIABLE
DEFINE L.FLAG AS AN INTEGER VARIABLE
DEFINE AIRFIELD AS AN INTEGER VARIABLE
DEFINE CRNT.BASE AS AN INTEGER VARIABLE
DEFINE HOME.BASE AS AN INTEGER VARIABLE
DEFINE DESTINATION.BASE AS AN INTEGER VARIABLE
DEFINE ALTERNATE.BASE AS AN INTEGER VARIABLE
DEFINE DESIGN AS AN ALPHA VARIABLE
DEFINE A.HOME.BASE AS AN ALPHA VARIABLE
DEFINE A.DESTINATION.BASE AS AN ALPHA VARIABLE
DEFINE A.ALTERNATE.BASE AS AN ALPHA VARIABLE
DEFINE A.1.MSN.AREA AS AN ALPHA VARIABLE
DEFINE A.2.MSN.AREA AS AN ALPHA VARIABLE
DEFINE A.3.MSN.AREA AS AN ALPHA VARIABLE
DEFINE A.CURRENT.BASE AS AN ALPHA VARIABLE
DEFINE LAST.MSN.AREA AS AN INTEGER VARIABLE
DEFINE NAME AS AN ALPHA VARIABLE
DEFINE MSN.1.AREA AS AN INTEGER VARIABLE
DEFINE MSN.2.AREA AS AN INTEGER VARIABLE
DEFINE MSN.3.AREA AS AN INTEGER VARIABLE
DEFINE 1.R.T.DESIGN AS AN ALPHA VARIABLE
DEFINE 2.R.T.DESIGN AS AN ALPHA VARIABLE
DEFINE 3.R.T.DESIGN AS AN ALPHA VARIABLE
```

```
 DEFINE 1.MAINT.SUP.CAP AS AN ALPHA VARIABLE
 DEFINE 2.MAINT.SUP.CAP AS AN ALPHA VARIABLE
 DEFINE 3.MAINT.SUP.CAP AS AN ALPHA VARIABLE
PRIORITY ORDER IS RELEASE.RUNWAY
BREAK LANDING TIES BY LOW FUEL2
 BREAK ARRIVAL.AT.IAF TIES BY LOW FUEL1
ACCUMULATE TO.TTL.HALF.HOUR AS THE PERIODIC NUMBER,
           TO.MAX.HALF.HOUR AS THE PERIODIC MAXIMUM,
           TO.MIN.HALF.HOUR AS THE PERIODIC MINIMUM,
           TO.AVG.HALF.HOUR AS THE PERIODIC MEAN
 OF N.TAKEOFF.QUEUE
ACCUMULATE OV.TTL.HALF.HOUR AS THE PERIODIC NUMBER,
           OV.MAX.HALF.HOUR AS THE PERIODIC MAXIMUM,
           OV.MIN.HALF.HOUR AS THE PERIODIC MINIMUM,
           OV.AVG.HALF.HOUR AS THE PERIODIC MEAN
 OF N.OVERHEAD.QUEUE
ACCUMULATE RE.TTL.HALF.HOUR AS THE PERIODIC NUMBER,
           RE.MAX.HALF.HOUR AS THE PERIODIC MAXIMUM,
           RE.MIN.HALF.HOUR AS THE PERICDIC MINIMUM,
           RE.AVG.HALF.HOUR AS THE PERIODIC MEAN
 OF N.RECTANGULAR.QUEUE
ACCUMULATE RD.TTL.HALF.HOUR AS THE PERIODIC NUMBER,
           RD.MAX.HALF.HOUR AS THE PERIODIC MAXIMUM,
           RD.MIN.HALF.HOUR AS THE PERIODIC MINIMUM,
           RD.AVG.HALF.HOUR AS THE PERIODIC MEAN
 OF N.RADAR.QUEUE
ACCUMULATE IA.TTL.HALF.HOUR AS THE PERIODIC NUMBER,
           IA.MAX.HALF.HOUR AS THE PERIODIC MAXIMUM,
           IA.MIN.HALF.HOUR AS THE PERIODIC MINIMUM,
           IA.AVG.HALF.HOUR AS THE PERIODIC MEAN
 OF N.IAF.QUEUE
ACCUMULATE MAX.TAKEOFF AS THE MAXIMUM, MEAN.TAKEOFF AS THE MEAN,
   VAR.TAKEOFF AS THE VARIANCE OF N.TAKEOFF.QUEUE
ACCUMULATE MAX.RECTANGULAR AS THE MAXIMUM, MEAN.RECTANGULAR AS THE MEAN,
   VAR.RECTANGULAR AS THE VARIANCE OF N.RECTANGULAR.QUEUE
ACCUMULATE MAX.IAF AS THE MAXIMUM, MEAN.IAF AS THE MEAN,
   VAR.IAF AS THE VARIANCE OF N.IAF.QUEUE
ACCUMULATE MAX.OVERHEAD AS THE MAXIMUM, MEAN.OVERHEAD AS THE MEAN,
   VAR.OVERHEAD AS THE VARIANCE OF N.OVERHEAD.QUEUE
ACCUMULATE MAX.RADAR AS THE MAXIMUM, MEAN.RADAR AS THE MEAN,
   VAR.RADAR AS THE VARIANCE OF N.RADAR.QUEUE
 END
```

```
MAIN
 LET LINES.V = 80
 LET DAY = 18
 LET N.BASE= 15
 CREATE EVERY BASE
 SKIP 2 LINES
 FOR EACH BASE
   DO
   READ
   NAME(BASE),
   X.POS(BASE),
   Y.POS(BASE),
   TRANSIENT.PARKING.SPACES(BASE),
   1.MAINT.SUP.CAP(BASE),
   2.MAINT.SUP.CAP(BASE),
   3.MAINT.SUP.CAP(BASE),
   MAXIRADAR.QUEUE(BASE),
   MAXIRECTANGULAR.QUEUE(BASE),
   MAXIOVERHEAD.QUEUE(BASE)
 PRINT 1 LINE WITH
   NAME(BASE),
   X.POS(BASE),
   Y.POS(BASE),
   TRANSIENT.PARKING.SPACES(BASE),
   1.MAINT.SUP.CAP(BASE),
   2.MAINT.SUP.CAP(BASE),
   3.MAINT.SUP.CAP(BASE),
   MAXIRADAR.QUEUE(BASE),
   MAXIRECTANGULAR.QUEUE(BASE) AND
   MAXIOVERHEAD.QUEUE(BASE)
 AS FOLLOWS
 *** *** *** *      * * *              * * *
 LET AVAIL.NO.TPS(BASE) = TRANSIENT.PARKING.SPACES(BASE)
  LOOP
 LET N.MSSN.AREA = 30
 CREATE EVERY MSSN.AREA
 SKIP 2 LINES
 FOR EACH MSSN.AREA
 DO
   READ
   X.COOR(MSSN.AREA),
   Y.COOR(MSSN.AREA)
   PRINT 1 LINE WITH MSSN.AREA, X.COOR(MSSN.AREA) AND Y.COOR(MSSN.AREA)
   AS FOLLOWS
   **    ***  ***
 LOOP
 START SIMULATION
 END
```

```
EVENT TAKEOFF GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 DEFINE Z.1HRS.TIME AS AN INTEGER VARIABLE
 DEFINE Z.2HRS.TIME AS AN INTEGER VARIABLE
 DEFINE Z.3HRS.TIME AS AN INTEGER VARIABLE
 DEFINE Z.1MINS.TIME AS AN INTEGER VARIABLE
 DEFINE Z.2MINS.TIME AS AN INTEGER VARIABLE
 DEFINE Z.3MINS.TIME AS AN INTEGER VARIABLE
 IF EVENT IS EXTERNAL
 CREATE AN AIRCRAFT CALLED PLANE
   READ
     TYPE(PLANE),
     DESIGN(PLANE),
     TAIL.NUMBER(PLANE),
     A.HOME.BASE(PLANE),
     A.DESTINATION.BASE(PLANE),
     A.ALTERNATE.BASE(PLANE),
     MISSION.TIME.REMAINING(PLANE),
     MSN.TYPE(PLANE),
     MSN.STAGE(PLANE),
     TRANSITION.TIME.REMAINING(PLANE),
     TOTAL.FUEL.REMAINING.TIME(PLANE),
     IAF.APPROACHES(PLANE),
     MISSED.APPROACHES(PLANE),
     RADAR.APPROACHES(PLANE),
     VISUAL.APPROACHES(PLANE),
     T.G.LANDINGS(PLANE),
     A.1.MSN.AREA(PLANE),
     M.1.AREA.TIME(PLANE),
     1.RCVR.TNKR.TAIL.NO(PLANE),
     1.ONLOAD.OFFLOAD(PLANE),
     1.MSN.OPTION(PLANE),
     1.DURATION(PLANE),
     A.2.MSN.AREA(PLANE),
     M.2.AREA.TIME(PLANE),
     2.RCVR.TNKR.TAIL.NO(PLANE),
     2.ONLOAD.OFFLOAD(PLANE),
     2.MSN.OPTION(PLANE),
     2.DURATION(PLANE),
     A.3.MSN.AREA(PLANE),
     M.3.AREA.TIME(PLANE),
     3.RCVR.TNKR.TAIL.NO(PLANE),
     3.ONLOAD.OFFLOAD(PLANE),
     3.MSN.OPTION(PLANE),
     3.DURATION(PLANE)
 CALL CONVERSION GIVEN PLANE
 LET BASE = HOME.BASE(PLANE)
 LET CRNT.BASE(PLANE) = BASE
```

```
LET SCH.IAF.APPRCACHES(PLANE)=IAF.APPROACHES(PLANE)
LET SCH.RADAR.APPROACHES(PLANE)=RADAR.APPROACHES(PLANE)
LET SCH.VISUAL.APPROACHES(PLANE)=VISUAL.APPROACHES(PLANE)
LET SCH.MISSED.APPROACHES(PLANE)=MISSED.APPROACHES(PLANE)
LET SCH.T.G.LANDINGS(PLANE)=T.G.LANDINGS(PLANE)
LET MISSION.TIME.REMAINING(PLANE) =
    MISSION.TIME.REMAINING(PLANE)* 60./1440.
LET TRANSITION.TIME.REMAINING(PLANE)=
    TRANSITION.TIME.REMAINING(PLANE)/1440.
LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
    TOTAL.FUEL.REMAINING.TIME(PLANE)* 60./1440.
LET Z.1TIME = M.1.AREA.TIME(PLANE)/100.
LET Z.2TIME = M.2.AREA.TIME(PLANE)/100.
LET Z.3TIME = M.3.AREA.TIME(PLANE)/100.
LET Z.1HRS.TIME = TRUNC.F(Z.1TIME)
LET Z.2HRS.TIME = TRUNC.F(Z.2TIME)
LET Z.3HRS.TIME = TRUNC.F(Z.3TIME)
LET Z.1MINS.TIME = (Z.1TIME- Z.1HRS.TIME)*100.
LET Z.2MINS.TIME = (Z.2TIME- Z.2HRS.TIME)*100.
LET Z.3MINS.TIME = (Z.3TIME- Z.3HRS.TIME)*100.
LET ZULU.1TIME = (Z.1HRS.TIME*60. + Z.1MINS.TIME)/1440.
LET ZULU.2TIME = (Z.2HRS.TIME*60. + Z.2MINS.TIME)/1440.
LET ZULU.3TIME = (Z.3HRS.TIME*60. + Z.3MINS.TIME)/1440.
LET M.1.AREA.TIME(PLANE) = ZULU.1TIME
LET M.2.AREA.TIME(PLANE) = ZULU.2TIME
LET M.3.AREA.TIME(PLANE) = ZULU.3TIME
LET 1.ONLOAD.OFFLOAD(PLANE) = 1.ONLOAD.OFFLOAD(PLANE) / 1440.
LET 2.ONLOAD.OFFLOAD(PLANE) = 2.ONLOAD.OFFLOAD(PLANE) / 1440.
LET 3.ONLOAD.OFFLOAD(PLANE) = 3.ONLOAD.OFFLOAD(PLANE) / 1440.
LET 1.DURATION(PLANE) = 1.DURATION(PLANE) / 1440.
LET 2.DURATION(PLANE) = 2.DURATION(PLANE) / 1440.
LET 3.DURATION(PLANE) = 3.DURATION(PLANE) / 1440.
    IF A.HOME.BASE(PLANE)= "BAB" OR
       A.HOME.BASE(PLANE) = "MHR" OR
       A.HOME.BASE(PLANE) = "SUU" OR
       A.HOME.BASE(PLANE) = "MER" OR
       A.HOME.BASE(PLANE) = "RIV"
     RESCHEDULE A TAKEOFF GIVEN PLANE IN 8 HOURS
   ELSE
     IF A.HOME.BASE(PLANE) = "SKA" OR
        A.HOME.BASE(PLANE) = "RCA"
       RESCHEDULE A TAKEOFF GIVEN PLANE IN 7 HOURS
     ELSE
       RESCHEDULE A TAKEOFF GIVEN PLANE IN 6 HOURS
     REGARDLESS
    REGARDLESS
GO TO RTRN
ELSE
```

```
IF PLANE IS IN A TAKEOFF.QUEUE
GO TO  CNTL
ELSE
LET TIM1.IN.TAKQUEUE(PLANE)= TIME.V
ADD 1 TO TO.COUNT
IF TO.COUNT = 1
SCHEDULE A HALF.FOUR.STATISTICS IN 30 MINUTES
ELSE REGARDLESS
FILE PLANE IN TAKEOFF.QUEUE(HOME.BASE(PLANE))
'CNTL'
LET BASE = HOME.BASE(PLANE)
LET CRNT.BASE(PLANE) = BASE
IF TIME.V - TIM1.IN.TAKQUEUE(PLANE) <(7./1440.)
   GO TO NOWT
ELSE
   FOR EACH LANDING IN EV.S(I.LANDING)
   WITH CRNT.BASE(PLANE4(LANDING)) = BASE
   FIND THE FIRST CASE
   IF NONE GO TO TAK
   ELSE
     IF TIME.A(LANDING) - TIME.V > (1./1440.)
   GO TO TAK
   ELSE
       IF RADAR.APPROACHES(PLANE4(LANDING)) <= 0 AND
          VISUAL.APPROACHES(PLANE4(LANDING)) <= 0 AND
          T.G.LANDINGS(PLANE4(LANDING)) <= 0 OR
          MISSION.TIME.REMAINING(PLANE4(LANDING)) < (-5./1440.)
      RESCHEDULE A TAKEOFF GIVEN PLANE AT TIME.A(LANDING)+(1./1440.)
      ELSE
      SUBTRACT 1 FROM MISSED.APPROACHES(PLANE4(LANDING))
      ADD 1 TO A.MISSED(PLANE4(LANDING))
      ADD 1 TO TTL.MISSED.APPROACHES(BASE)
      SCHEDULE A DEPARTURE.POINT GIVEN PLANE4(LANDING)
      AT TIME.A(LANDING) + (1./1440.)
      CANCEL THE LANDING
'TAK'
         IF RUNWAY(BASE) = BUSY
         ADD 1 TO NO.TAKEOFF.CONFLICTS(BASE)
         RESCHEDULE A TAKEOFF GIVEN PLANE IN 1 MINUTES
         GO TO RTRN
         ELSE
IF RUNWAY(HOME.BASE(PLANE)) = CLOSED
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
 PRINT 1 LINE WITH DESIGN(PLANE), TAIL.NUMBER(PLANE), A.HOME.BASE(PLANE)
    AND GMT(PLANE)
   AS FOLLOWS
   TAKEOFF CANX FOR * ****  BASE = ****  TIME(ZULU) = ****
```

```
    REMOVE THE PLANE FROM THE TAKEOFF.QUEUE(BASE)
    DESTROY THE AIRCRAFT CALLED PLANE
  GO TO RTRN
  ELSE
        LET LKT(PLANE) = TIME.V
        LET X.LAST.POS(PLANE) = X.POS(BASE)
        LET Y.LAST.POS(PLANE) = Y.POS(BASE)
  CALL TO.DECISION.RULES GIVEN PLANE
        LET RUNWAY(BASE) = BUSY
        LET TYME = TIME.V
        CALL CURRENT.TIME GIVEN PLANE AND TYME
         LET Z.TAKEOFF(PLANE) = GMT(PLANE)
         LET L.TO(PLANE) = LOCAL.TIME(PLANE)
         ADD 1 TO TTL.TAKEOFFS(BASE)
         IF (MSN.TYPE(PLANE)=3 OR MSN.TYPE(PLANE)=5)
            AND TYPE(PLANE) IS NOT EQUAL TO FTR.TYPE
         SUBTRACT 1 FROM TRANSIENT.PARKING.SPACES(DESTINATION.BASE(PLANE))
         ADD 1 TO TRANSIENT.PARKING.SPACES(HOME.BASE(PLANE))
         ELSE REGARDLESS
         REMOVE THE PLANE FROM THE TAKEOFF.QUEUE(BASE)
         LET TIM1.IN.TAKQUEUE(PLANE)=TIME.V-TIM1.IN.TAKQUEUE(PLANE)
         SCHEDULE A RELEASE.RUNWAY GIVEN BASE IN 1 MINUTES
         SCHEDULE A DEPARTURE.POINT GIVEN PLANE IN 1 MINUTES
         GO TO RTRN
  'NOWT'
    FOR EACH LANDING IN EV.S(I.LANDING)
    WITH CRNT.BASE(PLANE4(LANDING)) = BASE
    FIND THE FIRST CASE
    IF NONE GO TO TAK
    ELSE
     IF TIME.A(LANDING) - TIME.V > (1./1440.)
  GO TO TAK
  ELSE
         RESCHEDULE A TAKEOFF GIVEN PLANE AT TIME.A(LANDING)+(1./1440.)
  REGARDLESS
  'RTRN'
 RETURN
  END
```

```
EVENT DEPARTURE.POINT GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 LET BASE = CRNT.BASE(PLANE)
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
IF PLANE IS IN A RADAR.QUEUE
   REMOVE THE PLANE FROM THE RADAR.QUEUE(BASE)
   LET TIM4.IN.RADQUEUE(PLANE) = TIME.V - TIM4.IN.RADQUEUE(PLANE)
   LET TRANSITION.TIME.REMAINING(PLANE) =
         TRANSITION.TIME.REMAINING(PLANE) - TIM4.IN.RADQUEUE(PLANE)
   LET MISSION.TIME.REMAINING(PLANE) = MISSION.TIME.REMAINING(PLANE)
         - TIM4.IN.RADQUEUE(PLANE)
   LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
         TOTAL.FUEL.REMAINING.TIME(PLANE)-TIM4.IN.RADQUEUE(PLANE)
ELSE
    IF PLANE IS IN AN OVERHEAD.QUEUE
     REMOVE THE PLANE FROM THE OVERHEAD.QUEUE(BASE)
     LET TIM2.IN.OVRQUEUE(PLANE) = TIME.V - TIM2.IN.OVRQUEUE(PLANE)
     LET TRANSITION.TIME.REMAINING(PLANE) =
           TRANSITION.TIME.REMAINING(PLANE) - TIM2.IN.OVRQUEUE(PLANE)
     LET MISSION.TIME.REMAINING(PLANE) =
           MISSION.TIME.REMAINING(PLANE) - TIM2.IN.OVRQUEUE(PLANE)
     LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
           TOTAL.FUEL.REMAINING.TIME(PLANE)-TIM2.IN.OVRQUEUE(PLANE)
   ELSE
       IF PLANE IS IN A RECTANGULAR.QUEUE
         REMOVE THE PLANE FROM THE RECTANGULAR.QUEUE(BASE)
         LET TIM3.IN.RECQUEUE(PLANE) = TIME.V - TIM3.IN.RECQUEUE(PLANE)
         LET TRANSITION.TIME.REMAINING(PLANE)=
               TRANSITION.TIME.REMAINING(PLANE)-TIM3.IN.RECQUEUE(PLANE)
         LET MISSION.TIME.REMAINING(PLANE) =
         MISSION.TIME.REMAINING(PLANE) - TIM3.IN.RECQUEUE(PLANE)
         LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
         TOTAL.FUEL.REMAINING.TIME(PLANE)-TIM3.IN.RECQUEUE(PLANE)
       ELSE
           IF PLANE IS IN AN IAF.QUEUE
             REMOVE THE PLANE FROM THE IAF.QUEUE(BASE)
             LET TIM5.IN.IAFQUEUE(PLANE) = TIME.V-TIM5.IN.IAFQUEUE(PLANE)
             LET TRANSITION.TIME.REMAINING(PLANE) =
             TRANSITION.TIME.REMAINING(PLANE)-TIM5.IN.IAFQUEUE(PLANE)
             LET MISSION.TIME.REMAINING(PLANE) =
             MISSION.TIME.REMAINING(PLANE)-TIM5.IN.IAFQUEUE(PLANE)
             LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
             TOTAL.FUEL.REMAINING.TIME(PLANE)-TIM5.IN.IAFQUEUE(PLANE)
           ELSE
               IF MSN.STAGE(PLANE) IS NOT EQUAL TO JUST.TOOKOFF
               LET TRANSITION.TIME.REMAINING(PLANE) =
               TRANSITION.TIME.REMAINING(PLANE) - 9./1440.
```

```
                  LET MISSICN.TIME.REMAINING(PLANE) =
                  MISSION.TIME.REMAINING(PLANE) - 9./1440.
                  LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
                  TOTAL.FUEL.REMAINING.TIME(PLANE) - 9./1440.
                  GO TO SKIP
                  ELSE
              REGARDLESS
          REGARDLESS
 REGARDLESS
REGARDLESS
 LET MISSION.TIME.REMAINING(PLANE)=
     MISSION.TIME.REMAINING(PLANE)- 1./1440.
 LET TOTAL.FUEL.REMAINING.TIME(PLANE)=
     TOTAL.FUEL.REMAINING.TIME(PLANE)- 1./1440.
 'SKIP'
 IF MSN.STAGE(PLANE) = JUST.TOOKOFF
   OR EMER.DIVERT.FLAG(PLANE) = DIVERT
  SCHEDULE AN ENROUTE GIVEN PLANE NOW
  GO TO RTN
 ELSE
 IF MSN.STAGE(PLANE) = OTHER.BASE.PATTERN AND
   (MSN.TYPE(PLANE)=4 OR MSN.TYPE(PLANE)=2)
   AND TRANSITION.TIME.REMAINING(PLANE) < 0
   LET DESTINATION.BASE(PLANE) = HOME.BASE(PLANE)
   LET MSN.TYPE(PLANE) = 1
   SCHEDULE AN ENROUTE GIVEN PLANE NOW
   GO TO RTN
   ELSE
  'TEST'
   IF TYPE(PLANE) EQUALS FTR.TYPE, GO TO FTR
   ELSE
     IF RADAR.APPROACHES(PLANE) = 0
     AND N.RECTANGULAR.QUEUE(BASE)<MAXIRECTANGULAR.QUEUE(BASE)
     FILE PLANE IN RECTANGULAR.QUEUE(BASE)
    SUBTRACT 1 FROM VISUAL.APPROACHES(PLANE)
    ADD 1 TO A.VISUALS(PLANE)
 ADD 1 TO TTL.RECTANGULAR(BASE)
   LET TIM3.IN.RECQUEUE(PLANE)= TIME.V
   SCHEDULE A LANDING GIVEN PLANE
 AND TOTAL.FUEL.REMAINING.TIME(PLANE)  IN
 (N.RECTANGULAR.QUEUE(BASE)*1.+4.) MINUTES
     ELSE
      IF N.RADAR.QUEUE(BASE)<MAXIRADAR.QUEUE(BASE)
     FILE PLANE IN RADAR.QUEUE(BASE)
     SUBTRACT 1 FROM RADAR.APPROACHES(PLANE)
    ADD 1 TO A.RADARS(PLANE)
 ADD 1 TO TTL.RADAR(BASE)
   LET TIM4.IN.RADQUEUE(PLANE)= TIME.V
```

```
   SCHEDULE A LANDING GIVEN PLANE
AND TOTAL.FUEL.REMAINING.TIME(PLANE)
IN (N.RADAR.QUEUE(BASE)*3.+10.) MINUTES
     ELSE
    SCHEDULE AN ARRIVAL.AT.IAF GIVEN PLANE
AND TOTAL.FUEL.REMAINING.TIME(PLANE)
IN 20 MINUTES
    LET MISSION.TIME.REMAINING(PLANE)=MISSION.TIME.REMAINING(PLANE)
    - 20./1440.
    LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
    TOTAL.FUEL.REMAINING.TIME(PLANE) - 20./1440.
    LET TRANSITION.TIME.REMAINING(PLANE) =
    TRANSITION.TIME.REMAINING(PLANE) - 20./1440.
  REGARDLESS
 REGARDLESS
 GO TO RTN
'FTR'
  IF RADAR.APPROACHES(PLANE)=0 AND
  N.OVERHEAD.QUEUE(BASE) < MAX(OVERHEAD.QUEUE(BASE)
  FILE PLANE IN OVERHEAD.QUEUE(BASE)
  SUBTRACT 1 FROM VISUAL.APPROACHES(PLANE)
   ADD 1 TO A.VISUALS(PLANE)
ADD 1 TO TTL.OVERHEAD(BASE)
LET TIM2.IN.OVRQUEUE(PLANE) = TIME.V
SCHEDULE A LANDING GIVEN PLANE
AND TOTAL.FUEL.REMAINING.TIME(PLANE)
IN (N.OVERHEAD.QUEUE(BASE)*1.+3.) MINUTES
  ELSE
    IF N.RADAR.QUEUE(BASE)<MAX(RADAR.QUEUE(BASE)
     FILE PLANE IN RADAR.QUEUE(BASE)
     SUBTRACT 1 FROM RADAR.APPROACHES(PLANE)
   ADD 1 TO A.RADARS(PLANE)
ADD 1 TO TTL.RADAR(BASE)
  LET TIM4.IN.RADQUEUE(PLANE) = TIME.V
  SCHEDULE A LANDING GIVEN PLANE
AND TOTAL.FUEL.REMAINING.TIME(PLANE)
IN (N.RADAR.QUEUE(BASE)*3.+10.) MINUTES
    ELSE
    SCHEDULE A LANDING GIVEN PLANE
AND TOTAL.FUEL.REMAINING.TIME(PLANE)
IN 9 MINUTES
  REGARDLESS
 REGARDLESS
'RTN'
RETURN
END
```

```
EVENT ENROUTE GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 LET LKT(PLANE) = TIME.V
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
 IF PLANE IS IN A LOW.LEVEL.ROUTE
 REMOVE THE PLANE FROM THE LOW.LEVEL.ROUTE(LAST.MSN.AREA(PLANE))
 ELSE
 IF PLANE IS IN A REFUELING.TRACK
 REMOVE THE PLANE FROM THE REFUELING.TRACK(LAST.MSN.AREA(PLANE))
 ELSE
 IF PLANE IS IN A DELAY.AREA
 REMOVE THE PLANE FROM THE DELAY.AREA(LAST.MSN.AREA(PLANE))
 ELSE
 REGARDLESS
 REGARDLESS
 REGARDLESS
 IF EMER.DIVERT.FLAG(PLANE) = DIVERT
 CALL DEP.PT.DECISION.RULES GIVEN PLANE
 LET EMER.DIVERT.FLAG(PLANE) = 0
 GO TO RETURN
 ELSE
 LET X1 = X.LAST.POS(PLANE)
 LET Y1 = Y.LAST.POS(PLANE)
 IF MSN.STAGE(PLANE) = JUST.TOOKOFF AND
   (MSN.TYPE(PLANE)=3 OR MSN.TYPE(PLANE)=4)
 OR MSN.STAGE(PLANE) = OTHER.BASE.PATTERN
 'DEST'
 LET X2 = X.POS(DESTINATION.BASE(PLANE))
 LET Y2 = Y.POS(DESTINATION.BASE(PLANE))
 LET D = DISTANCE(X1,X2,Y1,Y2)
 LET MISSION.TIME.REMAINING(PLANE)=MISSION.TIME.REMAINING(PLANE)-D/1440
 LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
   TOTAL.FUEL.REMAINING.TIME(PLANE) - D/1440
 SCHEDULE AN ARRIVAL.AT.IAF GIVEN PLANE IN D MINUTES
 GO TO RETURN
 ELSE
     IF MSN.STAGE(PLANE) = MSN.COMPLETE AND
        MSN.2.AREA(PLANE) = 0 AND
        MSN.3.AREA(PLANE) = 0
     LET TYME = TIME.V
     CALL CURRENT.TIME GIVEN PLANE AND TYME
     LET 1.EXIT(PLANE) = GMT(PLANE)
     GO TO DEST
     ELSE
        IF MSN.STAGE(PLANE) = MSN.COMPLETE AND
          MSN.3.AREA(PLANE) = 0
        LET TYME = TIME.V
```

```
      CALL CURRENT.TIME GIVEN PLANE AND TYME
      LET 2.EXIT(PLANE) = GMT(PLANE)
      GO TO DEST
      ELSE
         IF MSN.STAGE(PLANE) = MSN.COMPLETE
         LET TYME = TIME.V
         CALL CURRENT.TIME GIVEN PLANE AND TYME
         LET 3.EXIT(PLANE) = GMT(PLANE)
         GO TO DEST
         ELSE
IF 1.MSN.OPTION(PLANE) > 0
GO TO SCHED1
ELSE
IF 2.MSN.OPTION(PLANE) > 0
GO TO SCHED2
ELSE
GO TO SCHED3
'SCHED1'
IF (DAY+M.1.AREA.TIME(PLANE)) < TIME.V
LET MISSION.TIME.REMAINING(PLANE) =
MISSION.TIME.REMAINING(PLANE)-(M.1.AREA.TIME(PLANE)+DAY+1-TIME.V)
LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
TOTAL.FUEL.REMAINING.TIME(PLANE) -
(M.1.AREA.TIME(PLANE)+DAY+1-TIME.V)
SCHEDULE A .MISSION GIVEN PLANE AT (M.1.AREA.TIME(PLANE)+DAY+1)
GO TO RETURN
ELSE
LET MISSION.TIME.REMAINING(PLANE) =
MISSION.TIME.REMAINING(PLANE)-(M.1.AREA.TIME(PLANE)+DAY-TIME.V)
LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
TOTAL.FUEL.REMAINING.TIME(PLANE) -
(M.1.AREA.TIME(PLANE)+DAY-TIME.V)
SCHEDULE A .MISSION GIVEN PLANE AT (M.1.AREA.TIME(PLANE)+DAY)
GO TO RETURN
'SCHED2'
LET TYME=TIME.V
CALL CURRENT.TIME GIVEN PLANE AND TYME
LET 1.EXIT(PLANE)=GMT(PLANE)
IF (DAY+M.2.AREA.TIME(PLANE)) < TIME.V
LET MISSION.TIME.REMAINING(PLANE) =
MISSION.TIME.REMAINING(PLANE)-(M.2.AREA.TIME(PLANE)+DAY+1-TIME.V)
LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
TOTAL.FUEL.REMAINING.TIME(PLANE) -
(M.2.AREA.TIME(PLANE)+DAY+1-TIME.V)
SCHEDULE A .MISSION GIVEN PLANE AT (M.2.AREA.TIME(PLANE)+DAY+1)
GO TO RETURN
ELSE
LET MISSION.TIME.REMAINING(PLANE) =
```

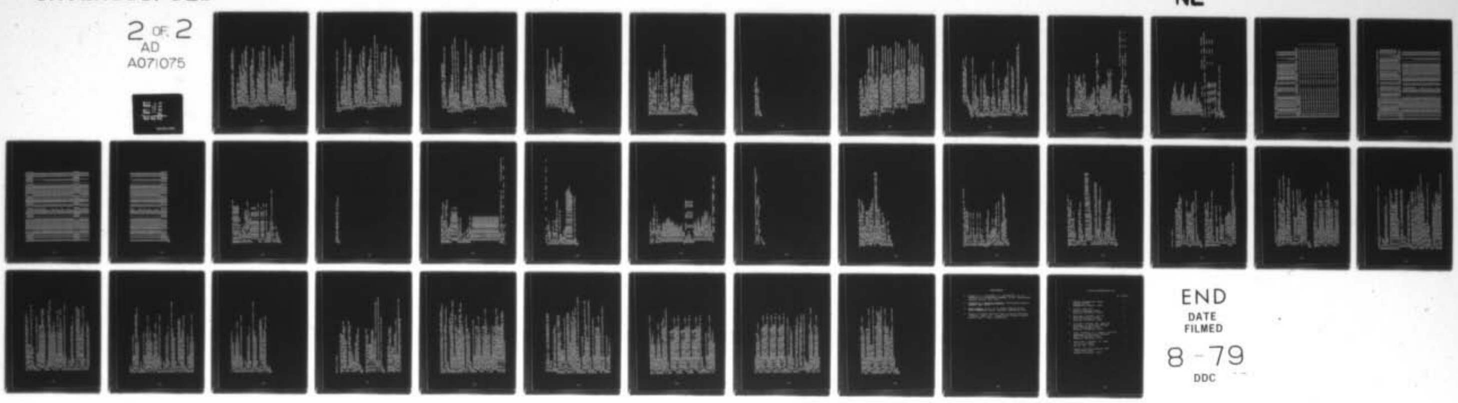
```
MISSION.TIME.REMAINING(PLANE)-(M.2.AREA.TIME(PLANE)+DAY-TIME.V)
LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
TOTAL.FUEL.REMAINING.TIME(PLANE) -
(M.2.AREA.TIME(PLANE)+DAY-TIME.V)
SCHEDULE A .MISSION GIVEN PLANE AT (M.2.AREA.TIME(PLANE)+DAY)
GO TO RETURN
'SCHED3'
LET TYME=TIME.V
CALL CURRENT.TIME GIVEN PLANE AND TYME
LET 2.EXIT(PLANE)=GMT(PLANE)
IF (DAY+M.3.AREA.TIME(PLANE)) < TIME.V
LET MISSION.TIME.REMAINING(PLANE) =
MISSION.TIME.REMAINING(PLANE)-(M.3.AREA.TIME(PLANE)+DAY+1-TIME.V)
LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
TOTAL.FUEL.REMAINING.TIME(PLANE) -
(M.3.AREA.TIME(PLANE)+DAY+1-TIME.V)
SCHEDULE A .MISSION GIVEN PLANE AT (M.3.AREA.TIME(PLANE)+DAY+1)
GO TO RETURN
ELSE
LET MISSION.TIME.REMAINING(PLANE) =
MISSION.TIME.REMAINING(PLANE)-(M.3.AREA.TIME(PLANE)+DAY-TIME.V)
LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
TOTAL.FUEL.REMAINING.TIME(PLANE) -
(M.3.AREA.TIME(PLANE)+DAY-TIME.V)
SCHEDULE A .MISSION GIVEN PLANE AT (M.3.AREA.TIME(PLANE)+DAY)
'RETURN'
RETURN
END
```

```
EVENT .MISSION GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 LET LKT(PLANE) = TIME.V
 IF 1.MSN.OPTION(PLANE) > 0
 LET MSSN.AREA = MSN.1.AREA(PLANE)
 LET X.LAST.POS(PLANE) = X.COOR(MSSN.AREA)
 LET Y.LAST.POS(PLANE) = Y.COOR(MSSN.AREA)
 LET LAST.MSN.AREA(PLANE) = MSSN.AREA
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
 LET 1.ENTRY(PLANE) = GMT(PLANE)
 GO TO MSN1
 ELSE
    IF 2.MSN.OPTION(PLANE) > 0
    LET MSSN.AREA = MSN.2.AREA(PLANE)
    LET X.LAST.POS(PLANE) = X.COOR(MSSN.AREA)
    LET Y.LAST.POS(PLANE) = Y.COOR(MSSN.AREA)
    LET LAST.MSN.AREA(PLANE) = MSSN.AREA
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
    LET 2.ENTRY(PLANE) = GMT(PLANE)
    GO TO MSN2
    ELSE
       LET MSSN.AREA = MSN.3.AREA(PLANE)
       LET X.LAST.POS(PLANE) = X.COOR(MSSN.AREA)
       LET Y.LAST.POS(PLANE) = Y.COOR(MSSN.AREA)
       LET LAST.MSN.AREA(PLANE) = MSSN.AREA
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
       LET 3.ENTRY(PLANE) = GMT(PLANE)
       GO TO MSN3
 'MSN1'
 IF 1.MSN.OPTION(PLANE) = 3
 LET 1.MSN.OPTION(PLANE) = 0
 FILE PLANE IN DELAY.AREA(MSSN.AREA)
 LET MISSION.TIME.REMAINING(PLANE)=MISSION.TIME.REMAINING(PLANE) -
  1.DURATION(PLANE)
 LET TOTAL.FUEL.REMAINING.TIME(PLANE)=TOTAL.FUEL.REMAINING.TIME(PLANE)-
  1.DURATION(PLANE)
 SCHEDULE AN ENROUTE GIVEN PLANE IN 1.DURATION(PLANE) DAYS
 GO TO RETN
 ELSE
    IF 1.MSN.OPTION(PLANE) = 2
    LET 1.MSN.OPTION(PLANE) = 0
    FILE PLANE IN LOW.LEVEL.ROUTE(MSSN.AREA)
    LET MISSION.TIME.REMAINING(PLANE) = MISSION.TIME.REMAINING(PLANE) -
     1.DURATION(PLANE)
    LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
```

AD-A071 075 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 17/7
A DEVELOPMENTAL COMPUTER MODEL FOR INVESTIGATIONS OF AIR TRAFFI--ETC(U)
MAR 79 J T MALOKAS, A P PEDERSON

UNCLASSIFIED NL

2 OF 2
AD
A071075



END
DATE
FILMED
8-79
DDC

```
  TOTAL.FUEL.REMAINING.TIME(PLANE) - 1.DURATION(PLANE)
 SCHEDULE AN ENROUTE GIVEN PLANE IN 1.DURATION(PLANE) DAYS
 GO TO RETN
 ELSE
   LET 1.MSN.OPTION(PLANE) = 0
   IF DESIGN(PLANE) = "K"
   FILE PLANE IN REFUELING.TRACK(MSSN.AREA)
   LET MISSION.TIME.REMAINING(PLANE) =
    MISSION.TIME.REMAINING(PLANE) - 1.DURATION(PLANE)
   LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
    TOTAL.FUEL.REMAINING.TIME(PLANE) - 1.DURATION(PLANE)
   SCHEDULE AN ENROUTE GIVEN PLANE IN 1.DURATION(PLANE) DAYS
   GO TO RETN
   ELSE
 FILE PLANE IN REFUELING.TRACK(MSSN.AREA)
 FOR EACH AIRCRAFT IN REFUELING.TRACK(MSSN.AREA)
   WITH TAIL.NUMBER(AIRCRAFT) = 1.RCVR.TNKR.TAIL.NO(PLANE)
 FIND THE FIRST CASE
   IF NONE
   LET 1.ONLOAD.OFFLOAD(PLANE) = 0
   LET MISSION.TIME.REMAINING(PLANE) =
     MISSION.TIME.REMAINING(PLANE) - 1.DURATION(PLANE)
   LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
     TOTAL.FUEL.REMAINING.TIME(PLANE)-1.DURATION(PLANE)
   SCHEDULE AN ENROUTE GIVEN PLANE IN 1.DURATION(PLANE) DAYS
   GO TO RETN
   ELSE
   LET MISSION.TIME.REMAINING(PLANE) =
     MISSION.TIME.REMAINING(PLANE)+1.ONLOAD.OFFLOAD(PLANE)
     - 1.DURATION(PLANE)
   LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
     TOTAL.FUEL.REMAINING.TIME(PLANE)
     + 1.ONLOAD.OFFLOAD(PLANE) - 1.DURATION(PLANE)
   LET TOTAL.FUEL.REMAINING.TIME(AIRCRAFT) =
    TOTAL.FUEL.REMAINING.TIME(AIRCRAFT) -
    1.ONLOAD.OFFLOAD(PLANE)
   LET 1.R.T.DESIGN(PLANE) = DESIGN(AIRCRAFT)
   LET 1.R.T.DESIGN(AIRCRAFT) = DESIGN(PLANE)
   SCHEDULE AN ENROUTE GIVEN PLANE IN 1.DURATION(PLANE) DAYS
   GO TO RETN
'MSN2'
IF 2.MSN.OPTION(PLANE) = 3
LET 2.MSN.OPTION(PLANE) = 0
FILE PLANE IN DELAY.AREA(MSSN.AREA)
LET MISSION.TIME.REMAINING(PLANE)=MISSION.TIME.REMAINING(PLANE) -
  2.DURATION(PLANE)
LET TOTAL.FUEL.REMAINING.TIME(PLANE)=TOTAL.FUEL.REMAINING.TIME(PLANE)-
  2.DURATION(PLANE)
```

```
SCHEDULE AN ENROUTE GIVEN PLANE IN 2.DURATION(PLANE) DAYS
GO TO RETN
ELSE
   IF 2.MSN.OPTION(PLANE) = 2
   LET 2.MSN.OPTION(PLANE) = 0
   FILE PLANE IN LOW.LEVEL.ROUTE(MSSN.AREA)
   LET MISSION.TIME.REMAINING(PLANE) = MISSION.TIME.REMAINING(PLANE) -
    2.DURATION(PLANE)
   LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
    TOTAL.FUEL.REMAINING.TIME(PLANE) - 2.DURATION(PLANE)
   SCHEDULE AN ENROUTE GIVEN PLANE IN 2.DURATION(PLANE) DAYS
   GO TO RETN
   ELSE
      LET 2.MSN.OPTION(PLANE) = 0
      IF DESIGN(PLANE) = "K"
      FILE PLANE IN REFUELING.TRACK(MSSN.AREA)
      LET MISSION.TIME.REMAINING(PLANE) =
        MISSION.TIME.REMAINING(PLANE) - 2.DURATION(PLANE)
      LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
        TOTAL.FUEL.REMAINING.TIME(PLANE) - 2.DURATION(PLANE)
      SCHEDULE AN ENROUTE GIVEN PLANE IN 2.DURATION(PLANE) DAYS
      GO TO RETN
      ELSE
   FILE PLANE IN REFUELING.TRACK(MSSN.AREA)
   FOR EACH AIRCRAFT IN REFUELING.TRACK(MSSN.AREA)
     WITH TAIL.NUMBER(AIRCRAFT) = 2.RCVR.TNKR.TAIL.NO(PLANE)
   FIND THE FIRST CASE
      IF NONE
      PRINT 1 LINE WITH TIME.V,A.2.MSN.AREA(PLANE) AND TAIL.NUMBER(PLANE)
      AS FOLLOWS
      AT **.****** NO TANKER FOUND IN AREA *** FOR AIRCRAFT ****.
      LET 2.ONLOAD.OFFLOAD(PLANE) = 0
      LET MISSION.TIME.REMAINING(PLANE) =
         MISSION.TIME.REMAINING(PLANE) - 2.DURATION(PLANE)
      LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
         TOTAL.FUEL.REMAINING.TIME(PLANE)-2.DURATION(PLANE)
      SCHEDULE AN ENROUTE GIVEN PLANE IN 2.DURATION(PLANE) DAYS
      GO TO RETN
      ELSE
      LET MISSION.TIME.REMAINING(PLANE) =
         MISSION.TIME.REMAINING(PLANE)+2.ONLOAD.OFFLOAD(PLANE)
         - 2.DURATION(PLANE)
      LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
         TOTAL.FUEL.REMAINING.TIME(PLANE)
         + 2.ONLOAD.OFFLOAD(PLANE) - 2.DURATION(PLANE)
      LET TOTAL.FUEL.REMAINING.TIME(AIRCRAFT) =
        TOTAL.FUEL.REMAINING.TIME(AIRCRAFT) -
        2.ONLOAD.OFFLOAD(PLANE)
```

```
     LET 2.R.T.DESIGN(PLANE) = DESIGN(AIRCRAFT)
     LET 2.R.T.DESIGN(AIRCRAFT) = DESIGN(PLANE)
     SCHEDULE AN ENROUTE GIVEN PLANE IN 2.DURATION(PLANE) DAYS
     GO TO RETN
'MSN3'
IF 3.MSN.OPTION(PLANE) = 3
LET 3.MSN.OPTION(PLANE) = 0
FILE PLANE IN DELAY.AREA(MSSN.AREA)
LET MISSION.TIMF.REMAINING(PLANE)=MISSICN.TIME.REMAINING(PLANE) -
 3.DURATION(PLANE)
LET TOTAL.FUEL.REMAINING.TIME(PLANE)=TOTAL.FUEL.REMAINING.TIME(PLANE)-
 3.DURATION(PLANE)
SCHEDULE AN ENROUTE GIVEN PLANE IN 3.DURATION(PLANE) DAYS
GO TO RETN
ELSE
   IF 3.MSN.OPTION(PLANE) = 2
   LET 3.MSN.OPTION(PLANE) = 0
   FILE PLANE IN LOW.LEVEL.ROUTE(MSSN.AREA)
   LET MISSION.TIME.REMAINING(PLANE) = MISSION.TIME.REMAINING(PLANE) -
    3.DURATION(PLANE)
   LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
    TOTAL.FUEL.REMAINING.TIME(PLANE) - 3.DURATION(PLANE)
   SCHEDULE AN ENROUTE GIVEN PLANE IN 3.DURATION(PLANE) DAYS
   GO TO RETN
   ELSE
      LET 3.MSN.OPTION(PLANE) = 0
      IF DESIGN(PLANE) = "K"
      FILE PLANE IN REFUELING.TRACK(MSSN.AREA)
      LET MISSION.TIME.REMAINING(PLANE) =
       MISSION.TIME.REMAINING(PLANE) - 3.DURATION(PLANE)
      LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
       TOTAL.FUEL.REMAINING.TIME(PLANE) - 3.DURATION(PLANE)
      SCHEDULE AN ENROUTE GIVEN PLANE IN 3.DURATION(PLANE) DAYS
      GO TO RETN
      ELSE
   FILE PLANE IN REFUELING.TRACK(MSSN.AREA)
   FOR EACH AIRCRAFT IN REFUELING.TRACK(MSSN.AREA)
      WITH TAIL.NUMBER(AIRCRAFT) = 3.RCVR.TNKR.TAIL.NO(PLANE)
   FIND THE FIRST CASE
      IF NONE
      LET 3.ONLOAD.OFFLOAD(PLANE) = 0
      LET MISSION.TIME.REMAINING(PLANE) =
         MISSION.TIME.REMAINING(PLANE) - 3.DURATION(PLANE)
      LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
         TOTAL.FUEL.REMAINING.TIME(PLANE)-3.DURATION(PLANE)
      SCHEDULE AN ENROUTE GIVEN PLANE IN 3.DURATION(PLANE) DAYS
      GO TO RETN
      ELSE
```

```
        LET MISSION.TIME.REMAINING(PLANE) =
          MISSION.TIME.REMAINING(PLANE)+3.ONLOAD.OFFLOAD(PLANE)
          - 3.DURATION(PLANE)
        LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
          TOTAL.FUEL.REMAINING.TIME(PLANE)
          + 3.ONLOAD.OFFLOAD(PLANE) - 3.DURATION(PLANE)
        LET TOTAL.FUEL.REMAINING.TIME(AIRCRAFT) =
         TOTAL.FUEL.REMAINING.TIME(AIRCRAFT) -
         3.ONLOAD.OFFLOAD(PLANE)
        LET 3.R.T.DESIGN(PLANE) = DESIGN(AIRCRAFT)
        LET 3.R.T.DESIGN(AIRCRAFT) = DESIGN(PLANE)
        SCHEDULE AN ENROUTE GIVEN PLANE IN 3.DURATION(PLANE) DAYS
   'RETN'
   IF 1.MSN.OPTION(PLANE) = 0 AND
      2.MSN.OPTION(PLANE) = 0 AND
      3.MSN.OPTION(PLANE) = 0
   LET MSN.STAGE(PLANE) = MSN.COMPLETE
   ELSE
   REGARDLESS
  RETURN
  END
```

```
EVENT ARRIVAL.AT.IAF GIVEN PLANE AND FUEL1
 DEFINE PLANE AS AN INTEGER VARIABLE
 LET LKT(PLANE) = TIME.V
 LET BASE = DESTINATION.BASE(PLANE)
 IF BASE = HOME.BASE(PLANE)
 LET MSN.STAGE(PLANE) = HOMEBASE.PATTERN
 ELSE
 LET MSN.STAGE(PLANE) = OTHER.BASE.PATTERN
 REGARDLESS
 LET CRNT.BASE(PLANE) = BASE
 IF BASE = 15
 PRINT 1 LINE WITH DESIGN(PLANE) AND TAIL.NUMBER(PLANE) AS FOLLOWS
 AIRCRAFT * **** LANDED AT NEAREST SUITABLE AIRFIELD
 DESTROY THE AIRCRAFT CALLED PLANE
 GO TO RETURN
 ELSE
 LET X.LAST.POS(PLANE) = X.POS(BASE)
 LET Y.LAST.POS(PLANE) = Y.POS(BASE)
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
 IF IAF.APPROACHES(PLANE) > 0
 LET Z.IAF(PLANE) = GMT(PLANE)
 LET L.IAF(PLANE) = LOCAL.TIME(PLANE)
 ELSE
 REGARDLESS
  SCHEDULE A LANDING GIVEN PLANE
 AND TOTAL.FUEL.REMAINING.TIME(PLANE)
 IN (N.IAF.QUEUE(BASE)*6.+15.) MINUTES
  LET TIM5.IN.IAFQUEUE(PLANE) = TIME.V
  FILE THE PLANE IN THE IAF.QUEUE(BASE)
  ADD 1 TO A.IAFS(PLANE)
  SUBTRACT 1 FROM IAF.APPROACHES(PLANE)
 ADD 1 TO TTL.IAF(BASE)
 'RETURN'
RETURN
END
```

```
EVENT RELEASE.RUNWAY GIVEN FIELD
   DEFINE FIELD AS AN INTEGER VARIABLE
   LET RUNWAY(FIELD) = IDLE
RETURN
END
```

```
EVENT LANDING GIVEN PLANE AND FUEL2
 DEFINE PLANE AS AN INTEGER VARIABLE
 LET BASE = CRNT.BASE(PLANE)
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
IF PLANE IS IN A RADAR.QUEUE
  REMOVE THE PLANE FROM THE RADAR.QUEUE(BASE)
  LET TIM4.IN.RADQUEUE(PLANE) = TIME.V - TIM4.IN.RADQUEUE(PLANE)
  LET TRANSITION.TIME.REMAINING(PLANE) =
      TRANSITION.TIME.REMAINING(PLANE) - TIM4.IN.RADQUEUE(PLANE)
  LET MISSION.TIME.REMAINING(PLANE) = MISSION.TIME.REMAINING(PLANE)
      - TIM4.IN.RADQUEUE(PLANE)
  LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
      TOTAL.FUEL.REMAINING.TIME(PLANE)-TIM4.IN.RADQUEUE(PLANE)
ELSE
   IF PLANE IS IN AN OVERHEAD.QUEUE
    REMOVE THE PLANE FROM THE OVERHEAD.QUEUE(BASE)
    LET TIM2.IN.OVRQUEUE(PLANE) = TIME.V - TIM2.IN.OVRQUEUE(PLANE)
    LET TRANSITION.TIME.REMAINING(PLANE) =
        TRANSITION.TIME.REMAINING(PLANE) - TIM2.IN.OVRQUEUE(PLANE)
    LET MISSION.TIME.REMAINING(PLANE) =
        MISSION.TIME.REMAINING(PLANE) - TIM2.IN.OVRQUEUE(PLANE)
    LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
        TOTAL.FUEL.REMAINING.TIME(PLANE)-TIM2.IN.CVRQUEUE(PLANE)
  ELSE
      IF PLANE IS IN A RECTANGULAR.QUEUE
        REMOVE THE PLANE FROM THE RECTANGULAR.QUEUE(BASE)
        LET TIM3.IN.RECQUEUE(PLANE) = TIME.V - TIM3.IN.RECQUEUE(PLANE)
        LET TRANSITION.TIME.REMAINING(PLANE)=
            TRANSITION.TIME.REMAINING(PLANE)-TIM3.IN.RECQUEUE(PLANE)
        LET MISSION.TIME.REMAINING(PLANE) =
        MISSION.TIME.REMAINING(PLANE) - TIM3.IN.RECQUEUE(PLANE)
        LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
        TOTAL.FUEL.REMAINING.TIME(PLANE)-TIM3.IN.RECQUEUE(PLANE)
      ELSE
          IF PLANE IS IN AN IAF.QUEUE
            REMOVE THE PLANE FROM THE IAF.QUEUE(BASE)
            LET TIM5.IN.IAFQUEUE(PLANE) = TIME.V-TIM5.IN.IAFQUEUE(PLANE)
            LET TRANSITION.TIME.REMAINING(PLANE) =
            TRANSITION.TIME.REMAINING(PLANE)-TIM5.IN.IAFQUEUE(PLANE)
            LET MISSION.TIME.REMAINING(PLANE) =
            MISSION.TIME.REMAINING(PLANE)-TIM5.IN.IAFQUEUE(PLANE)
            LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
            TOTAL.FUEL.REMAINING.TIME(PLANE)-TIM5.IN.IAFQUEUE(PLANE)
            ELSE
            LET TRANSITION.TIME.REMAINING(PLANE) =
              TRANSITION.TIME.REMAINING(PLANE) - 9./1440.
            LET MISSION.TIME.REMAINING(PLANE) =
```

```
                MISSION.TIME.REMAINING(PLANE) - 9./1440.
              LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
                TOTAL.FUEL.REMAINING.TIME(PLANE) - 9./1440.
            REGARDLESS
        REGARDLESS
    REGARDLESS
REGARDLESS
FOR EACH LANDING IN EV.S(I.LANDING) WITH TIME.A(LANDING)
    - TIME.V <= 1./1440.
  AND CRNT.BASE(PLANE4(LANDING)) = BASE
FIND THE FIRST CASE
IF NONE GO TO APP
ELSE
 ADD 1 TO NO.LANDING.CONFLICTS(BASE)
HERE
FOR EACH LANDING IN EV.S(I.LANDING)
 WITH CRNT.BASE(PLANE4(LANDING)) = BASE
 AND L.FLAG(LANDING) = 0
 FIND THE FIRST CASE
 IF NONE GO TO ZERO
 ELSE
CANCEL THE LANDING
RESCHEDULE THE LANDING AT TIME.A(LANDING) + 1./1440.
LET L.FLAG(LANDING) = 1
JUMP BACK
 'ZERO'
 FOR EACH LANDING IN EV.S(I.LANDING)
 LET L.FLAG(LANDING) = 0
'APP'
 IF RUNWAY(BASE) = CLOSED
 ADD 1 TO A.MISSED(PLANE)
 ADD 1 TO TTL.MISSED.APPROACHES(BASE)
 LET EMER.DIVERT.FLAG(PLANE) = DIVERT
 SCHEDULE A DEPARTURE.POINT GIVEN PLANE IN 1 MINUTES
 GO TO RETURN
 ELSE
   IF MSN.STAGE(PLANE) = OTHER.BASE.PATTERN AND
     (MSN.TYPE(PLANE)=2 OR MSN.TYPE(PLANE)=4)
   GO TO T.G
   ELSE
IF RADAR.APPROACHES(PLANE)<=0 AND VISUAL.APPROACHES(PLANE)<=0 AND
 T.G.LANDINGS(PLANE)<=0 OR MISSION.TIME.REMAINING(PLANE)<(-15./1440.)
  GO TO FSL
ELSE
  'T.G'
  ADD 1 TO A.T.GS(PLANE)
  SUBTRACT 1 FROM T.G.LANDINGS(PLANE)
 ADD 1 TO TTL.T.G.LANDINGS(BASE)
```

```
 ADD 1 TO TTL.LANDINGS(BASE)
  SCHEDULE A DEPARTURE.POINT GIVEN PLANE
IN 1 MINUTES
 GO TO RET
'FSL'
 LET TYME = TIME.V
 CALL CURRENT.TIME GIVEN PLANE AND TYME
 LET Z.LANDING(PLANE) = GMT(PLANE)
 LET L.LNDG(PLANE) = LOCAL.TIME(PLANE)
 LET A.CURRENT.BASE(PLANE) = NAME(BASE)
 ADD 1 TO TTL.FSL.LANDINGS(BASE)
 ADD 1 TO TTL.LANDINGS(BASE)
 IF BASE IS NOT EQUAL TO HOME.BASE(PLANE)
 ADD 1 TO TTL.NO.TRANSIENTS(BASE)
 ELSE REGARDLESS
 PRINT 2 LINES WITH
   DESIGN(PLANE),
   TAIL.NUMBER(PLANE),
   A.HOME.BASE(PLANE),
   A.CURRENT.BASE(PLANE) AND
   A.ALTERNATE.BASE(PLANE)
 AS FOLLOWS
 DESIGN   TAIL.NUMBER   HCME.BASE CURRENT.BASE  ALTERNATE.BASE
   **        ****           ****        ****          ****
 PRINT 3 DOUBLE LINES WITH
   SCH.IAF.APPROACHES(PLANE),
   SCH.VISUAL.APPROACHES(PLANE),
   SCH.RADAR.APPROACHES(PLANE),
   SCH.MISSED.APPROACHES(PLANE),
   SCH.T.G.LANDINGS(PLANE),
  (TOTAL.FUEL.REMAINING.TIME(PLANE) * 1440.),
  (MISSION.TIME.REMAINING(PLANE) * 1440.),
  (TRANSITION.TIME.REMAINING(PLANE) * 1440.),
   A.IAFS(PLANE),
   A.VISUALS(PLANE),
   A.RADARS(PLANE),
   A.MISSED(PLANE) AND
   A.T.GS(PLANE)
   AS FOLLOWS
 TIMES (MINUTES REMAINING):                         ATC ACTIVITIES: IAFS VISUALS RAD
ARS MISSED.APPROACHES TOUCH.AND.GOS
 TOTAL.FLYING.TIME MISSION.TIME TRANSITION.TIME SCHEDULED          **     **       *
*           **                **
      ****.**          ****.**          ****.**    ACCOMPLISHED       **     **       *
*           **                **
 PRINT 5 DOUBLE LINES WITH
   A.1.MSN.AREA(PLANE),
   1.ENTRY(PLANE),
```

```
    1.EXIT(PLANE),
    1.R.T.DESIGN(PLANE),
    1.RCVR.TNKR.TAIL.NO(PLANE),
    (1.ONLOAD.OFFLOAD(PLANE)*1440),
    A.2.MSN.AREA(PLANE),
    2.ENTRY(PLANE),
    2.EXIT(PLANE),
    2.R.T.DESIGN(PLANE),
    2.RCVR.TNKR.TAIL.NO(PLANE),
    (2.ONLOAD.OFFLOAD(PLANE)*1440),
    Z.TAKEOFF(PLANE),
    Z.IAF(PLANE),
    Z.LANDING(PLANE),
    A.3.MSN.AREA(PLANE),
    3.ENTRY(PLANE),
    3.EXIT(PLANE),
    3.R.T.DESIGN(PLANE),
    3.RCVR.TNKR.TAIL.NO(PLANE),
    (3.ONLOAD.OFFLOAD(PLANE)*1440),
    L.TO(PLANE),
    L.IAF(PLANE) AND
    L.LNDG(PLANE)
    AS FOLLOWS
 MISSION               ENTRY EXIT                          RECEIVER/       TAIL     OFFLOAD/

 ACTIVITIES    AREA  TIME   TIME  REFUELING:  TANKER DESIGN  NUMBER   ONLOAD(MINS)

     1         ****  ****   ****                   **         ****      ****
 EVENT TIMES:  TAKEOFF IAF  LANDING
     2         ****  ****   ****                   **         ****      ****
    ZULU         ****  ****   ****
     3         ****  ****   ****                   **         ****      ****
    LOCAL        ****  ****   ****
DESTROY THE AIRCRAFT CALLED PLANE
 'RET'
  LET RUNWAY(BASE) = BUSY
SCHEDULE A RELEASE.RUNWAY GIVEN BASE IN 1 MINUTES
 'RETURN'
RETURN
END
```

```
ROUTINE CONVERSION GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 IF A.HOME.BASE(PLANE) = "SKA"  LET HOME.BASE(PLANE) =  1  ELSE
 IF A.HOME.BASE(PLANE) = "BAB"  LET HOME.BASE(PLANE) =  2  ELSE
 IF A.HOME.BASE(PLANE) = "MHR"  LET HOME.BASE(PLANE) =  3  ELSE
 IF A.HOME.BASE(PLANE) = "SUU"  LET HOME.BASE(PLANE) =  4  ELSE
 IF A.HOME.BASE(PLANE) = "MER"  LET HOME.BASE(PLANE) =  5  ELSE
 IF A.HOME.BASE(PLANE) = "RIV"  LET HOME.BASE(PLANE) =  6  ELSE
 IF A.HOME.BASE(PLANE) = "RCA"  LET HOME.BASE(PLANE) =  7  ELSE
 IF A.HOME.BASE(PLANE) = "MIB"  LET HOME.BASE(PLANE) =  8  ELSE
 IF A.HOME.BASE(PLANE) = "RDR"  LET HOME.BASE(PLANE) =  9  ELSE
 IF A.HOME.BASE(PLANE) = "OFF"  LET HOME.BASE(PLANE) = 10  ELSE
 IF A.HOME.BASE(PLANE) = "IAB"  LET HOME.BASE(PLANE) = 11  ELSE
 IF A.HOME.BASE(PLANE) = "LTS"  LET HOME.BASE(PLANE) = 12  ELSE
 IF A.HOME.BASE(PLANE) = "DYS"  LET HOME.BASE(PLANE) = 13  ELSE
 IF A.HOME.BASE(PLANE) = "FWH"  LET HOME.BASE(PLANE) = 14  ELSE
 REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
 REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
 REGARDLESS REGARDLESS
 IF A.DESTINATION.BASE(PLANE) = "SKA" LET DESTINATION.BASE(PLANE) =  1
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "BAB" LET DESTINATION.BASE(PLANE) =  2
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "MHR" LET DESTINATION.BASE(PLANE) =  3
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "SUU" LET DESTINATION.BASE(PLANE) =  4
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "MER" LET DESTINATION.BASE(PLANE) =  5
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "RIV" LET DESTINATION.BASE(PLANE) =  6
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "RCA" LET DESTINATION.BASE(PLANE) =  7
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "MIB" LET DESTINATION.BASE(PLANE) =  8
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "RDR" LET DESTINATION.BASE(PLANE) =  9
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "OFF" LET DESTINATION.BASE(PLANE) = 10
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "IAB" LET DESTINATION.BASE(PLANE) = 11
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "LTS" LET DESTINATION.BASE(PLANE) = 12
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "DYS" LET DESTINATION.BASE(PLANE) = 13
 ELSE
 IF A.DESTINATION.BASE(PLANE) = "FWH" LET DESTINATION.BASE(PLANE) = 14
 ELSE
 REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
```

```
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS
IF A.ALTERNATE.BASE(PLANE)="SKA" LET ALTERNATE.BASE(PLANE)= 1 ELSE
IF A.ALTERNATE.BASE(PLANE)="BAB" LET ALTERNATE.BASE(PLANE)= 2 ELSE
IF A.ALTERNATE.BASE(PLANE)="MHR" LET ALTERNATE.BASE(PLANE)= 3 ELSE
IF A.ALTERNATE.BASE(PLANE)="SUU" LET ALTERNATE.BASE(PLANE)= 4 ELSE
IF A.ALTERNATE.BASE(PLANE)="MER" LET ALTERNATE.BASE(PLANE)= 5 ELSE
IF A.ALTEPNATE.BASE(PLANE)="RIV" LET ALTERNATE.BASE(PLANE)= 6 ELSE
IF A.ALTERNATE.BASE(PLANE)="RCA" LET ALTERNATE.BASE(PLANE)= 7 ELSE
IF A.ALTERNATE.BASE(PLANE)="MIB" LET ALTERNATE.BASE(PLANE)= 8 ELSE
IF A.ALTERNATE.BASE(PLANE)="RDR" LET ALTERNATE.BASE(PLANE)= 9 ELSE
IF A.ALTERNATE.BASE(PLANE)="OFF" LET ALTERNATE.BASE(PLANE)=10 ELSE
IF A.ALTERNATE.BASE(PLANE)="IAB" LET ALTERNATE.BASE(PLANE)=11 ELSE
IF A.ALTERNATE.BASE(PLANE)="LTS" LET ALTERNATE.BASE(PLANE)=12 ELSE
IF A.ALTERNATE.BASE(PLANE)="DYS" LET ALTERNATE.BASE(PLANE)=13 ELSE
IF A.ALTERNATE.BASE(PLANE)="FWH" LET ALTERNATE.BASE(PLANE)=14
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS
IF A.1.MSN.AREA(PLANE) = "OB1"   LET MSN.1.AREA(PLANE) =  1  ELSE
IF A.1.MSN.AREA(PLANE) = "OB10"  LET MSN.1.AREA(PLANE) =  2  ELSE
IF A.1.MSN.AREA(PLANE) = "OB18"  LET MSN.1.AREA(PLANE) =  3  ELSE
IF A.1.MSN.AREA(PLANE) = "OB20"  LET MSN.1.AREA(PLANE) =  4  ELSE
IF A.1.MSN.AREA(PLANE) = "OB24"  LET MSN.1.AREA(PLANE) =  5  ELSE
IF A.1.MSN.AREA(PLANE) = "OB30"  LET MSN.1.AREA(PLANE) =  6  ELSE
IF A.1.MSN.AREA(PLANE) = "OB46"  LET MSN.1.AREA(PLANE) =  7  ELSE
IF A.1.MSN.AREA(PLANE) = "OB95"  LET MSN.1.AREA(PLANE) =  8  ELSE
IF A.1.MSN.AREA(PLANE) = "OB29"  LET MSN.1.AREA(PLANE) =  9  ELSE
IF A.1.MSN.AREA(PLANE) = "OB97"  LET MSN.1.AREA(PLANE) = 10  ELSE
IF A.1.MSN.AREA(PLANE) = "OB9"   LET MSN.1.AREA(PLANE) = 11  ELSE
IF A.1.MSN.AREA(PLANE) = "A9"    LET MSN.1.AREA(PLANE) = 12  ELSE
IF A.1.MSN.AREA(PLANE) = "A92"   LET MSN.1.AREA(PLANE) = 13  ELSE
IF A.1.MSN.AREA(PLANE) = "A93"   LET MSN.1.AREA(PLANE) = 14  ELSE
IF A.1.MSN.AREA(PLANE) = "A916"  LET MSN.1.AREA(PLANE) = 15  ELSE
IF A.1.MSN.AREA(PLANE) = "A320"  LET MSN.1.AREA(PLANE) = 16  ELSE
IF A.1.MSN.AREA(PLANE) = "A5"    LET MSN.1.AREA(PLANE) = 17  ELSE
IF A.1.MSN.AREA(PLANE) = "A106"  LET MSN.1.AREA(PLANE) = 18  ELSE
IF A.1.MSN.AREA(PLANE) = "A28"   LET MSN.1.AREA(PLANE) = 19  ELSE
IF A.1.MSN.AREA(PLANE) = "A55"   LET MSN.1.AREA(PLANE) = 20  ELSE
IF A.1.MSN.AREA(PLANE) = "W92"   LET MSN.1.AREA(PLANE) = 21  ELSE
IF A.1.MSN.AREA(PLANE) = "W320"  LET MSN.1.AREA(PLANE) = 22  ELSE
IF A.1.MSN.AREA(PLANE) = "W28"   LET MSN.1.AREA(PLANE) = 23  ELSE
IF A.1.MSN.AREA(PLANE) = "W5"    LET MSN.1.AREA(PLANE) = 24  ELSE
IF A.1.MSN.AREA(PLANE) = "W93"   LET MSN.1.AREA(PLANE) = 25  ELSE
IF A.1.MSN.AREA(PLANE) = "W22"   LET MSN.1.AREA(PLANE) = 26  ELSE
IF A.1.MSN.AREA(PLANE) = "W319"  LET MSN.1.AREA(PLANE) = 27  ELSE
IF A.1.MSN.AREA(PLANE) = "W96"   LET MSN.1.AREA(PLANE) = 28  ELSE
IF A.1.MSN.AREA(PLANE) = "A96"   LET MSN.1.AREA(PLANE) = 29  ELSE
```

```
IF A.1.MSN.AREA(PLANE) = "A7"   LET MSN.1.AREA(PLANE) = 30  ELSE
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
IF A.2.MSN.AREA(PLANE) = "OB1"  LET MSN.2.AREA(PLANE) =  1  ELSE
IF A.2.MSN.AREA(PLANE) = "OB10" LET MSN.2.AREA(PLANE) =  2  ELSE
IF A.2.MSN.AREA(PLANE) = "OB18" LET MSN.2.AREA(PLANE) =  3  ELSE
IF A.2.MSN.AREA(PLANE) = "OB20" LET MSN.2.AREA(PLANE) =  4  ELSE
IF A.2.MSN.AREA(PLANE) = "OB24" LET MSN.2.AREA(PLANE) =  5  ELSE
IF A.2.MSN.AREA(PLANE) = "OB30" LET MSN.2.AREA(PLANE) =  6  ELSE
IF A.2.MSN.AREA(PLANE) = "OB46" LET MSN.2.AREA(PLANE) =  7  ELSE
IF A.2.MSN.AREA(PLANE) = "OB95" LET MSN.2.AREA(PLANE) =  8  ELSE
IF A.2.MSN.AREA(PLANE) = "OB29" LET MSN.2.AREA(PLANE) =  9  ELSE
IF A.2.MSN.AREA(PLANE) = "OB97" LET MSN.2.AREA(PLANE) = 10  ELSE
IF A.2.MSN.AREA(PLANE) = "OB9"  LET MSN.2.AREA(PLANE) = 11  ELSE
IF A.2.MSN.AREA(PLANE) = "A9"   LET MSN.2.AREA(PLANE) = 12  ELSE
IF A.2.MSN.AREA(PLANE) = "A92"  LET MSN.2.AREA(PLANE) = 13  ELSE
IF A.2.MSN.AREA(PLANE) = "A93"  LET MSN.2.AREA(PLANE) = 14  ELSE
IF A.2.MSN.AREA(PLANE) = "A916" LET MSN.2.AREA(PLANE) = 15  ELSE
IF A.2.MSN.AREA(PLANE) = "A320" LET MSN.2.AREA(PLANE) = 16  ELSE
IF A.2.MSN.AREA(PLANE) = "A5"   LET MSN.2.AREA(PLANE) = 17  ELSE
IF A.2.MSN.AREA(PLANE) = "A106" LET MSN.2.AREA(PLANE) = 18  ELSE
IF A.2.MSN.AREA(PLANE) = "A28"  LET MSN.2.AREA(PLANE) = 19  ELSE
IF A.2.MSN.AREA(PLANE) = "A55"  LET MSN.2.AREA(PLANE) = 20  ELSE
IF A.2.MSN.AREA(PLANE) = "W92"  LET MSN.2.AREA(PLANE) = 21  ELSE
IF A.2.MSN.AREA(PLANE) = "W320" LET MSN.2.AREA(PLANE) = 22  ELSE
IF A.2.MSN.AREA(PLANE) = "W28"  LET MSN.2.AREA(PLANE) = 23  ELSE
IF A.2.MSN.AREA(PLANE) = "W5"   LET MSN.2.AREA(PLANE) = 24  ELSE
IF A.2.MSN.AREA(PLANE) = "W93"  LET MSN.2.AREA(PLANE) = 25  ELSE
IF A.2.MSN.AREA(PLANE) = "W22"  LET MSN.2.AREA(PLANE) = 26  ELSE
IF A.2.MSN.AREA(PLANE) = "W319" LET MSN.2.AREA(PLANE) = 27  ELSE
IF A.2.MSN.AREA(PLANE) = "W96"  LET MSN.2.AREA(PLANE) = 28  ELSE
IF A.2.MSN.AREA(PLANE) = "A96"  LET MSN.2.AREA(PLANE) = 29  ELSE
IF A.2.MSN.AREA(PLANE) = "A7"   LET MSN.2.AREA(PLANE) = 30  ELSE
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
IF A.3.MSN.AREA(PLANE) = "OB1"  LET MSN.3.AREA(PLANE) =  1  ELSE
IF A.3.MSN.AREA(PLANE) = "OB10" LET MSN.3.AREA(PLANE) =  2  ELSE
IF A.3.MSN.AREA(PLANE) = "OB18" LET MSN.3.AREA(PLANE) =  3  ELSE
IF A.3.MSN.AREA(PLANE) = "OB20" LET MSN.3.AREA(PLANE) =  4  ELSE
IF A.3.MSN.AREA(PLANE) = "OB24" LET MSN.3.AREA(PLANE) =  5  ELSE
IF A.3.MSN.AREA(PLANE) = "OB30" LET MSN.3.AREA(PLANE) =  6  ELSE
IF A.3.MSN.AREA(PLANE) = "OB46" LET MSN.3.AREA(PLANE) =  7  ELSE
```

```
IF A.3.MSN.AREA(PLANE) = "OB95" LET MSN.3.AREA(PLANE) =  8  ELSE
IF A.3.MSN.AREA(PLANE) = "OB29" LET MSN.3.AREA(PLANE) =  9  ELSE
IF A.3.MSN.AREA(PLANE) = "OB97" LET MSN.3.AREA(PLANE) = 10  ELSE
IF A.3.MSN.AREA(PLANE) = "OB9"  LET MSN.3.AREA(PLANE) = 11  ELSE
IF A.3.MSN.AREA(PLANE) = "A9"   LET MSN.3.AREA(PLANE) = 12  ELSE
IF A.3.MSN.AREA(PLANE) = "A92"  LET MSN.3.AREA(PLANE) = 13  ELSE
IF A.3.MSN.AREA(PLANE) = "A93"  LET MSN.3.AREA(PLANE) = 14  ELSE
IF A.3.MSN.AREA(PLANE) = "A916" LET MSN.3.AREA(PLANE) = 15  ELSE
IF A.3.MSN.AREA(PLANE) = "A320" LET MSN.3.AREA(PLANE) = 16  ELSE
IF A.3.MSN.AREA(PLANE) = "A5"   LET MSN.3.AREA(PLANE) = 17  ELSE
IF A.3.MSN.AREA(PLANE) = "A106" LET MSN.3.AREA(PLANE) = 18  ELSE
IF A.3.MSN.AREA(PLANE) = "A28"  LET MSN.3.AREA(PLANE) = 19  ELSE
IF A.3.MSN.AREA(PLANE) = "A55"  LET MSN.3.AREA(PLANE) = 20  ELSE
IF A.3.MSN.AREA(PLANE) = "W92"  LET MSN.3.AREA(PLANE) = 21  ELSE
IF A.3.MSN.AREA(PLANE) = "W320" LET MSN.3.AREA(PLANE) = 22  ELSE
IF A.3.MSN.AREA(PLANE) = "W28"  LET MSN.3.AREA(PLANE) = 23  ELSE
IF A.3.MSN.AREA(PLANE) = "W5"   LET MSN.3.AREA(PLANE) = 24  ELSE
IF A.3.MSN.AREA(PLANE) = "W93"  LET MSN.3.AREA(PLANE) = 25  ELSE
IF A.3.MSN.AREA(PLANE) = "W22"  LET MSN.3.AREA(PLANE) = 26  ELSE
IF A.3.MSN.AREA(PLANE) = "W319" LET MSN.3.AREA(PLANE) = 27  ELSE
IF A.3.MSN.AREA(PLANE) = "W96"  LET MSN.3.AREA(PLANE) = 28  ELSE
IF A.3.MSN.AREA(PLANE) = "A96"  LET MSN.3.AREA(PLANE) = 29  ELSE
IF A.3.MSN.AREA(PLANE) = "A7"   LET MSN.3.AREA(PLANE) = 30  ELSE
 REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
 REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
 REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
 REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
 REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS REGARDLESS
RETURN
END
```

```
ROUTINE CURRENT.TIME GIVEN PLANE AND TYME
 DEFINE PLANE AS AN INTEGER VARIABLE
 DEFINE ZHRS AS AN INTEGER VARIABLE
 DEFINE ZMINS AS AN INTEGER VARIABLE
 DEFINE L.DAY AS AN INTEGER VARIABLE
 LET ZNTIME = TYME - DAY
 IF ZNTIME >= 1
 LET DAY = DAY + 1
 ELSE REGARDLESS
 LET ZNTIME = (TYME-  DAY)*1440.
 LET ZHRS = TRUNC.F(ZNTIME/60.)
 LET ZMINS = TRUNC.F(ZNTIME - 60 * ZHRS)
 LET GMTIME = ZHRS*100 + ZMINS
 LET GMT(PLANE) = GMTIME
 IF A.HOME.BASE(PLANE) = "BAB"  OR
    A.HOME.BASE(PLANE) = "MHR"  OR
    A.HOME.BASE(PLANE) = "SUU"  OR
    A.HOME.BASE(PLANE) = "MER"  OR
    A.HOME.BASE(PLANE) = "RIV"
 LET LOCAL.TIME(PLANE) = GMTIME - 800
 ELSE
 IF A.HOME.BASE(PLANE) = "SKA"  OR
    A.HOME.BASE(PLANE) = "RCA"
 LET LOCAL.TIME(PLANE) = GMTIME - 700
 ELSE
 LET LOCAL.TIME(PLANE) = GMTIME - 600
 REGARDLESS
 REGARDLESS
 IF LOCAL.TIME(PLANE) < 0
  LET LOCAL.TIME(PLANE) = LOCAL.TIME(PLANE) + 2400
 LET L.DAY = DAY - 1
 ELSE
 LET L.DAY = DAY
 REGARDLESS
RETURN
END
```

```
 ROUTINE FOR DISTANCE (X1, X2, Y1, Y2)
 RETURN WITH SORT.F((X2-X1)**2 + (Y2-Y1)**2)
END
```

```
EVENT HALF.HOUR.STATISTICS
 DEFINE ACTIVITY AS AN  INTEGER VARIABLE
 LET ACTIVITY = N.BASE
 PRINT 1 LINE WITH TIME.V AS FOLLOWS
 HALF HOUR STATISTICS     TIME = **.******
 FOR EACH BASE
 DO THE FOLLOWING
 IF TO.TTL.HALF.HOUR(BASE) = 0
    AND  OV.TTL.HALF.HOUR(BASE) = 0
    AND  RE.TTL.HALF.HOUR(BASE) = 0
    AND  RD.TTL.HALF.HOUR(BASE) = 0
    AND  IA.TTL.HALF.HOUR(BASE) = 0
 SUBTRACT 1 FROM ACTIVITY
 IF ACTIVITY = 0
 ADD 1 TO END.SIMULATION
 GO TO END
 ELSE
 REGARDLESS
 LOOP
 FOR EACH BASE
 WITH BASE <= 14
 PRINT 3 DOUBLE LINES WITH
    NAME(BASE),
    TO.MAX.HALF.HOUR(BASE),
    OV.MAX.HALF.HOUR(BASE),
    RE.MAX.HALF.HOUR(BASE),
    RD.MAX.HALF.HOUR(BASE),
    IA.MAX.HALF.HOUR(BASE),
    TO.TTL.HALF.HOUR(BASE),
    OV.TTL.HALF.HOUR(BASE),
    RE.TTL.HALF.HOUR(BASE),
    RD.TTL.HALF.HOUR(BASE),
    IA.TTL.HALF.HOUR(BASE),
    TO.MIN.HALF.HOUR(BASE),
    OV.MIN.HALF.HOUR(BASE),
    RE.MIN.HALF.HOUR(BASE),
    RD.MIN.HALF.HOUR(BASE),
    IA.MIN.HALF.HOUR(BASE),
    TO.AVG.HALF.HOUR(BASE),
    OV.AVG.HALF.HOUR(BASE),
    RE.AVG.HALF.HOUR(BASE),
    RC.AVG.HALF.HOUR(BASE),
    IA.AVG.HALF.HOUR(BASE)
 AS FOLLOWS
 BASE=****  TAKEOFF OVERHEAD RECTANGULAR RADAR IAF        TAKEOFF OVERHEAD RECTA
NGULAR  RADAR   IAF
    MAXIMUM   ***        ***        ***        ***  ***  TOTAL   ***        ***        *
**        ***    ***
```

```
   MINIMUM    ***       ***        ***       ***  ***  MEAN   **.**    **.**     **
.**     **.**  **.**
 LET END.SIMULATION = 0
 GO TO RETURN
 'END'
 IF END.SIMULATION < 48
 SKIP 1 LINE
 PRINT 1 LINE AS FOLLOWS
 NO ACTIVITY IN BASE QUEUES FOR 30 MINUTES
 ELSE
   SCHEDULE A STOP.SIMULATION NOW
 REGARDLESS
 'RETURN'
 FOR EACH BASE
 DO THE FOLLOWING
 RESET THE PERIODIC TOTALS OF N.TAKEOFF.QUEUE(BASE)
 RESET THE PERIODIC TOTALS OF N.OVERHEAD.QUEUE(BASE)
 RESET THE PERIODIC TOTALS OF N.RECTANGULAR.QUEUE(BASE)
 RESET THE PERIODIC TOTALS OF N.RADAR.QUEUE(BASE)
 RESET THE PERIODIC TOTALS OF N.IAF.QUEUE(BASE)
 LOOP
 SCHEDULE A HALF.HOUR.STATISTICS IN 30 MINUTES
RETURN
END
```

```
EVENT STOP.SIMULATION
 SKIP 2 LINES
 PRINT 1 LINE AS FOLLOWS
          ACCUMULATED BASE STATISTICS
 SKIP 2 LINES
 FOR EACH BASE WITH BASE <= 14
 DO THE FOLLOWING
 PRINT 7 LINES WITH
  NAME(BASE),
  MAX.TAKEOFF(BASE),
  MEAN.TAKEOFF(BASE),
  VAR.TAKEOFF(BASE),
  MAX.RECTANGULAR(BASE),
  MEAN.RECTANGULAR(BASE),
  VAR.RECTANGULAR(BASE),
  MAX.OVERHEAD(BASE),
  MEAN.OVERHEAD(BASE),
  VAR.OVERHEAD(BASE),
  MAX.RADAR(BASE),
  MEAN.RADAR(BASE),
  VAR.RADAR(BASE),
  MAX.IAF(BASE),
  MEAN.IAF(BASE),
  AND VAR.IAF(BASE)
  AS FOLLOWS
BASE = ****
 QUEUES        MAXIMUM     MEAN      VARIANCE
 TAKEOFF         ***      ***.**      ***.**
 RECTANGULAR     ***      ***.**      ***.**
 OVERHEAD        ***      ***.**      ***.**
 RADAR           ***      ***.**      ***.**
 IAF             ***      ***.**      ***.**
 PRINT 6 LINES WITH
   TTL.TAKEOFFS(BASE),
   TTL.RADAR(BASE),
   TTL.RECTANGULAR(BASE),
   TTL.OVERHEAD(BASE),
   TTL.IAF(BASE),
   TTL.FSL.LANDINGS(BASE),
   TTL.T.G.LANDINGS(BASE),
   TTL.MISSED.APPROACHES(BASE),
   NO.TAKEOFF.CONFLICTS(BASE),
   NO.LANDING.CONFLICTS(BASE),
   AVAIL.NO.TPS(BASE),
   TTL.NO.TRANSIENTS(BASE)
   AS FOLLOWS
 TOTALS:  TAKEOFFS RADARS RECTANGULARS OVERHEADS   IAFS   FSLS
           *****.   *****.    *****.      *****.   *****. *****.
```

```
TOTAL.T.G.S TOTAL.MISSED.APPS NO.TAKEOFF.CONFLICTS NO.LANDING.CONFLICTS
    *****.          *****.          *****.                     *****.
AVAILABLE.PARKING.SPOTS  NUMBER.OF.TRANSIENTS
             ***                   ***
    LOOP
STOP
END
```

```
ROUTINE CLOSEST.BASE GIVEN X1 AND Y1 YIELDING
  NEAREST.BASE AND MIN.DISTANCE
 DEFINE NEAREST.BASE AS AN INTEGER VARIABLE
 DEFINE MIN.DISTANCE AS A REAL VARIABLE
  FOR EACH BASE WITH
  X1 NOT EQUAL TO X.POS(BASE) AND
  Y1 NOT EQUAL TO Y.POS(BASE) AND
  RUNWAY(BASE) NOT EQUAL TO CLOSED
 AND SEARCH.FLAG(BASE) NOT EQUAL TO 1
  COMPUTE MIN.DISTANCE AS THE MINIMUM OF
   DISTANCE(X1, X.POS(BASE), Y1, Y.POS(BASE))
 FOR EACH BASE WITH
   DISTANCE(X1,X.POS(BASE),Y1,Y.POS(BASE))>=(MIN.DISTANCE- 0.001) AND
   DISTANCE(X1,X.POS(BASE),Y1,Y.POS(BASE))<=(MIN.DISTANCE+ 0.001) AND
   RUNWAY(BASE) NOT EQUAL TO CLOSED
 AND SEARCH.FLAG(BASE) NOT EQUAL TO 1
 FIND THE FIRST CASE
 IF NONE PRINT 1 LINE AS FOLLOWS
 OOPS NO CLOSEST BASE FOUND
 LET NEAREST.BASE = 99
 ELSE
 LET NEAREST.BASE = BASE
 REGARDLESS
RETURN
END
```

```
ROUTINE POSITION GIVEN   X1,X2,Y1,Y2,D1 AND D YIELDING
 X.CURRENT.POS, Y.CURRENT.POS AND D2
 LET X.FACT = (D1*ABS.F(X2-X1))/D
 LET Y.FACT = (D1*ABS.F(Y2-Y1))/D
 IF X1 > X2
 LET X.CURRENT.POS = X1 - X.FACT
 ELSE
 LET X.CURRENT.POS = X1 + X.FACT
 REGARDLESS
 IF Y1 > Y2
 LET Y.CURRENT.POS = Y1 - Y.FACT
 ELSE
 LET Y.CURRENT.POS = Y1 + Y.FACT
 REGARDLESS
 LET D2 = D - D1
RETURN
END
EVENT CHANGE.RUNWAY.STATUS
 DEFINE FIELD AS AN ALPHA VARIABLE
 READ FIELD
  FOR EACH BASE
   WITH NAME(BASE) = FIELD
 DO THE FOLLOWING
 LET RUNWAY(BASE) = CLOSED
 PRINT 1 LINE WITH NAME(BASE) AND TIME.V
 AS FOLLOWS
 RUNWAY AT BASE = **** CLOSED AT **.******
 LOOP
 CALL AB.DECISION.RULES
 FOR EACH RELEASE.RUNWAY IN EV.S(I.RELEASE.RUNWAY)
   WITH NAME(AIRFIELD(RELEASE.RUNWAY)) = FIELD
 CANCEL THE RELEASE.RUNWAY
 RETURN
 END
```

```
ROUTINE NRST.SUPPORT.BASE GIVEN PLANE,X1 AND Y1
 DEFINE PLANE AS AN INTEGER VARIABLE
 DEFINE NEAREST.BASE AS AN INTEGER VARIABLE
 IF TYPE(PLANE) = FTR.TYPE
LET DESTINATION.BASE(PLANE) = 15
 GO TO RETURN
 ELSE
 'CALL'
 CALL CLOSEST.BASE GIVEN X1 AND Y1
   YIELDING NEAREST.BASE AND MIN.DISTANCE
 IF NEAREST.BASE = 99
 GO TO ZERO
 ELSE
 IF (DESIGN(PLANE) IS NOT EQUAL TO 1.MAINT.SUP.CAP(NEAREST.BASE) AND
      DESIGN(PLANE) IS NOT EQUAL TO 2.MAINT.SUP.CAP(NEAREST.BASE) AND
      DESIGN(PLANE) IS NOT EQUAL TO 3.MAINT.SUP.CAP(NEAREST.BASE)) OR
      TRANSIENT.PARKING.SPACES(NEAREST.BASE) <= 0
 LET SEARCH.FLAG(NEAREST.BASE) = 1
 GO TO CALL
 ELSE
 IF DESTINATION.BASE(PLANE) IS NOT EQUAL TO HOME.BASE(PLANE)
 ADD 1 TO TRANSIENT.PARKING.SPACES(DESTINATION.BASE(PLANE))
 ELSE REGARDLESS
 LET DESTINATION.BASE(PLANE) = NEAREST.BASE
 SUBTRACT 1 FROM TRANSIENT.PARKING.SPACES(NEAREST.BASE)
 GO TO RETURN
 'ZERO'
 FOR EACH BASE
 LET SEARCH.FLAG(BASE) = 0
 CALL CLOSEST.BASE GIVEN X1 AND Y1
   YIELDING NEAREST.BASE AND MIN.DISTANCE
 LET DESTINATION.BASE(PLANE) = NEAREST.BASE
 'RETURN'
 FOR EACH BASE
 LET SEARCH.FLAG(BASE) = 0
RETURN
END
```

STRATEGY 1 MODULE

```
ROUTINE TO.DECISION.RULES GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 DEFINE NEAREST.BASE AS AN INTEGER VARIABLE
 DEFINE X1,Y1, AND MIN.DISTANCE AS REAL VARIABLES
   IF RUNWAY(DESTINATION.BASE(PLANE)) = CLOSED
   GO TO ALT
   ELSE GO TO RTN
  'ALT'
   IF RUNWAY(ALTERNATE.BASE(PLANE)) = CLOSED
 LET X1 = X.POS(CRNT.BASE(PLANE))
 LET Y1 = Y.POS(CRNT.BASE(PLANE))
 CALL CLOSEST.BASE GIVEN X1 AND Y1
   YIELDING NEAREST.BASE AND MIN.DISTANCE
 LET DESTINATION.BASE(PLANE) = NEAREST.BASE
  ELSE
  LET DESTINATION.BASE(PLANE) = ALTERNATE.BASE(PLANE)
  REGARDLESS
  'RTN'
RETURN
END

ROUTINE DEP.PT.DECISION.RULES GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 DEFINE NEAREST.BASE AS AN INTEGER VARIABLE
 IF TYPE(PLANE) = FTR.TYPE
 LET DESTINATION.BASE(PLANE) = 15
 SCHEDULE AN ARRIVAL.AT.IAF GIVEN PLANE IN 30 MINUTES
 GO TO RET
 ELSE
 IF MSN.STAGE(PLANE) = HOMEBASE.PATTERN
 GO CHECK.ALTERNATE
 ELSE
   IF RUNWAY(HOME.BASE(PLANE)) = CLOSED
   LET MSN.TYPE(PLANE) = 3
   GO CHECK.ALTERNATE
   ELSE
   LET X1 = X.LAST.POS(PLANE)
   LET Y1 = Y.LAST.POS(PLANE)
 LET X2 = X.POS(HOME.BASE(PLANE))
 LET Y2 = Y.POS(HOME.BASE(PLANE))
 LET D = DISTANCE(X1,X2,Y1,Y2)
   SCHEDULE AN ARRIVAL.AT.IAF GIVEN PLANE IN D MINUTES
   LET MISSION.TIME.REMAINING(PLANE) = MISSION.TIME.REMAINING(PLANE) - D/ 1440.
   LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
```

```
      TOTAL.FUEL.REMAINING.TIME(PLANE) - D/ 1440.
    GO TO RET
  'CHECK.ALTERNATE'
   IF RUNWAY(ALTERNATE.BASE(PLANE)) = CLCSED
   CALL CLOSEST.BASE GIVEN X.LAST.POS(PLANE) AND Y.LAST.POS(PLANE)
    YIELDING NEAREST.BASE AND MIN.DISTANCE
    LET DESTINATION.BASE(PLANE) = NEAREST.BASE
    LET MISSION.TIME.REMAINING(PLANE) = MISSION.TIME.REMAINING(PLANE) -
      MIN.DISTANCE/1440
    LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
      TOTAL.FUEL.REMAINING.TIME(PLANE) - MIN.DISTANCE/1440
    SCHEDULE AN ARRIVAL.AT.IAF GIVEN PLANE IN MIN.CISTANCE MINUTES
    GO TO RET
   ELSE
    LET DESTINATION.BASE(PLANE) = ALTERNATE.BASE(PLANE)
    LET X1 = X.LAST.POS(PLANE)
    LET Y1 = Y.LAST.POS(PLANE)
    LET D = DISTANCE(X1 , X.POS(DESTINATION.BASE(PLANE)),
      Y1, Y.POS(DESTINATION.BASE(PLANE)))
    SCHEDULE AN ARRIVAL.AT.IAF GIVEN PLANE IN D MINUTES
   LET MISSION.TIME.REMAINING(PLANE) =
      MISSION.TIME.REMAINING(PLANE) - D/1440
    LET TOTAL.FUEL.REMAINING.TIME(PLANE) =
      TOTAL.FUEL.REMAINING.TIME(PLANE) - D/1440
  'RET'
RETURN
END


ROUTINE AB.DECISION.RULES
 DEFINE NEAREST.BASE AS AN INTEGER VARIABLE
 FOR EACH ARRIVAL.AT.IAF IN EV.S(I.ARRIVAL.AT.IAF)
 DO THE FOLLOWING
 IF (MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 2 OR
   MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 4) AND
   RUNWAY(HOME.BASE(PLANE3(ARRIVAL.AT.IAF))) = CLOSED
 LET MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 3
 ELSE REGARDLESS
 IF (MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 2 OR
   MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 4) AND
   RUNWAY(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF))) = CLOSED
 LET DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)) =
   HOME.BASE(PLANE3(ARRIVAL.AT.IAF))
 LET MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 1
 ELSE REGARDLESS
 IF (MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 1 OR
   MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 3 OR
   MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 5) AND
```

```
  RUNWAY(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF))) = CLOSED
GO TO CK.ALT1
ELSE
GO TO LOOP1
'CK.ALT1'
IF TYPE(PLANE3(ARRIVAL.AT.IAF))=FTR.TYPE
LET DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)) = 15
GO TO LOOP1
ELSE
IF RUNWAY(ALTERNATE.BASE(PLANE3(ARRIVAL.AT.IAF)))=CLOSED
LET D1 = (TIME.V-LKT(PLANE3(ARRIVAL.AT.IAF))) * 1440
LET X1 = X.LAST.POS(PLANE3(ARRIVAL.AT.IAF))
LET Y1 = Y.LAST.POS(PLANE3(ARRIVAL.AT.IAF))
LET X2 = X.POS(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)))
LET Y2 = Y.POS(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)))
LET D = DISTANCE(X1,X2,Y1,Y2)
  IF D1 = 0
  LET X.CURRENT.POS = X.LAST.POS(PLANE3(ARRIVAL.AT.IAF))
  LET Y.CURRENT.POS = Y.LAST.POS(PLANE3(ARRIVAL.AT.IAF))
  LET D2 = D
  GO TO PRNT
  ELSE
  CALL POSITION GIVEN X1,X2,Y1,Y2,D1 AND D
  YIELDING X.CURRENT.POS, Y.CURRENT.POS AND D2
'PRNT'
LET X.LAST.POS(PLANE3(ARRIVAL.AT.IAF)) = X.CURRENT.POS
LET Y.LAST.POS(PLANE3(ARRIVAL.AT.IAF)) = Y.CURRENT.POS
LET LKT(PLANE3(ARRIVAL.AT.IAF)) = TIME.V
CALL CLOSEST.BASE GIVEN X.CURRENT.POS AND Y.CURRENT.POS
  YIELDING NEAREST.BASE AND MIN.DISTANCE
LET DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)) = NEAREST.BASE
LET MISSION.TIME.REMAINING(PLANE3(ARRIVAL.AT.IAF)) =
MISSION.TIME.REMAINING(PLANE3(ARRIVAL.AT.IAF)) +(D2-MIN.DISTANCE)/1440
LET TOTAL.FUEL.REMAINING.TIME(PLANE3(ARRIVAL.AT.IAF)) =
TOTAL.FUEL.REMAINING.TIME(PLANE3(ARRIVAL.AT.IAF)) +
  (D2-MIN.DISTANCE)/1440
CANCEL THE ARRIVAL.AT.IAF
RESCHEDULE THE ARRIVAL.AT.IAF GIVEN PLANE3(ARRIVAL.AT.IAF) IN
  MIN.DISTANCE MINUTES
GO TO LOOP1
ELSE
LET DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)) =
  ALTERNATE.BASE(PLANE3(ARRIVAL.AT.IAF))
LET X2= X.POS(ALTERNATE.BASE(PLANE3(ARRIVAL.AT.IAF)))
LET Y2= Y.POS(ALTERNATE.BASE(PLANE3(ARRIVAL.AT.IAF)))
LET D = DISTANCE(X.CURRENT.POS,X2,Y.CURRENT.POS,Y2)
LET MISSION.TIME.REMAINING(PLANE3(ARRIVAL.AT.IAF)) =
MISSION.TIME.REMAINING(PLANE3(ARRIVAL.AT.IAF)) +(D2-MIN.DISTANCE)/1440
```

```
LET TOTAL.FUEL.REMAINING.TIME(PLANE3(ARRIVAL.AT.IAF)) =
TOTAL.FUEL.REMAINING.TIME(PLANE3(ARRIVAL.AT.IAF)) +
   (D2-MIN.DISTANCE)/1440
CANCEL THE ARRIVAL.AT.IAF
RESCHEDULE THE ARRIVAL.AT.IAF GIVEN PLANE3(ARRIVAL.AT.IAF) IN
   MIN.DISTANCE MINUTES
'LOOP1'
LOOP
FOR EACH DEPARTURE.POINT IN EV.S(I.DEPARTURE.POINT)
DO THE FOLLOWING
IF  (MSN.TYPE(PLANE2(DEPARTURE.POINT))=2 OR
   MSN.TYPE(PLANE2(DEPARTURE.POINT))=4) AND
   MSN.STAGE(PLANE2(DEPARTURE.POINT)) = OTHER.BASE.PATTERN
   AND RUNWAY(HOME.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 3
ELSE
REGARDLESS
IF  MSN.STAGE(PLANE2(DEPARTURE.POINT)) = JUST.TOOKOFF AND
     RUNWAY(DESTINATION.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED AND
   MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 2
LET MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 1
LET DESTINATION.BASE(PLANE2(DEPARTURE.POINT)) =
   HOME.BASE(PLANE2(DEPARTURE.POINT))
ELSE REGARDLESS
IF MSN.STAGE(PLANE2(DEPARTURE.POINT)) = JUST.TOOKOFF AND
   MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 4
   AND RUNWAY(HOME.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 3
ELSE REGARDLESS
IF RUNWAY(DESTINATION.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
GO TO CK.ALT2
ELSE
GO TO LOOP2
'CK.ALT2'
IF  TYPE(PLANE2(DEPARTURE.POINT))=FTR.TYPE
LET DESTINATION.BASE(PLANE2(DEPARTURE.POINT)) = 15
GO TO LOOP2
ELSE
IF RUNWAY(ALTERNATE.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET X1 = X.POS(DESTINATION.BASE(PLANE2(DEPARTURE.POINT)))
LET Y1 = Y.POS(DESTINATION.BASE(PLANE2(DEPARTURE.POINT)))
CALL CLOSEST.BASE GIVEN X1 AND Y1
   YIELDING NEAREST.BASE AND MIN.DISTANCE
LET DESTINATION.BASE(PLANE2(DEPARTURE.POINT)) = NEAREST.BASE
GO TO LOOP2
ELSE
LET DESTINATION.BASE(PLANE2(DEPARTURE.POINT)) =
   ALTERNATE.BASE(PLANE2(DEPARTURE.POINT))
```

```
'LOOP2'
LOOP
FOR EACH LANDING IN EV.S(I.LANDING)
DO THE FOLLOWING
IF MSN.STAGE(PLANE4(LANDING)) = OTHER.BASE.PATTERN AND
 (MSN.TYPE(PLANE4(LANDING))=2 OR MSN.TYPE(PLANE4(LANDING))=4)
LET MSN.TYPE(PLANE4(LANDING)) = 3
ELSE
REGARDLESS
LOOP
FOR EACH .MISSION IN EV.S(I..MISSION)
DO THE FOLLOWING
IF MSN.TYPE(PLANE5(.MISSION)) = 2 AND
   RUNWAY(DESTINATION.BASE(PLANE5(.MISSION))) = CLOSED
LET MSN.TYPE(PLANE5(.MISSION)) = 1
LET DESTINATION.BASE(PLANE5(.MISSION)) =
   HOME.BASE(PLANE5(.MISSION))
ELSE REGARDLESS
IF MSN.TYPE(PLANE5(.MISSION)) = 2 AND
   RUNWAY(HOME.BASE(PLANE5(.MISSION))) = CLOSED
LET MSN.TYPE(PLANE5(.MISSION)) = 5
ELSE REGARDLESS
IF RUNWAY(DESTINATION.BASE(PLANE5(.MISSION))) = CLOSED
GO TO CK.ALT3
ELSE
GO TO LOOP3
'CK.ALT3'
IF TYPE(PLANE5(.MISSION))=FTR.TYPE
LET DESTINATION.BASE(PLANE5(.MISSION)) = 15
GO TO LOOP3
ELSE
IF RUNWAY(ALTERNATE.BASE(PLANE5(.MISSION))) IS NOT EQUAL TO CLOSED
LET DESTINATION.BASE(PLANE5(.MISSION)) =
 ALTERNATE.BASE(PLANE5(.MISSION))
GO TO LOOP3
   ELSE
LET X1 = X.POS(DESTINATION.BASE(PLANE5(.MISSION)))
LET Y1 = Y.POS(DESTINATION.BASE(PLANE5(.MISSION)))
CALL CLOSEST.BASE GIVEN X1 AND Y1
   YIELDING NEAREST.BASE AND MIN.DISTANCE
LET DESTINATION.BASE(PLANE5(.MISSION)) = NEAREST.BASE
'LOOP3'
LOOP
FOR EACH ENROUTE IN EV.S(I.ENROUTE)
DO THE FOLLOWING
IF MSN.TYPE(PLANE6(ENROUTE)) = 2
AND RUNWAY(DESTINATION.BASE(PLANE6(ENROUTE))) = CLOSED
LET MSN.TYPE(PLANE6(ENROUTE)) = 1
```

```
LET DESTINATION.BASE(PLANE6(ENROUTE)) =
  HOME.BASE(PLANE6(ENROUTE))
ELSE REGARDLESS
IF MSN.TYPE(PLANE6(ENROUTE)) = 2
AND RUNWAY(HOME.BASE(PLANE6(ENROUTE))) = CLOSED
LET MSN.TYPE(PLANE6(ENROUTE)) = 5
ELSE REGARDLESS
IF RUNWAY(DESTINATION.BASE(PLANE6(ENROUTE))) = CLOSED
GO TO CK.ALT4
ELSE
GO TO LOOP4
'CK.ALT4'
IF TYPE(PLANE6(ENROUTE))=FTR.TYPE
LET DESTINATION.BASE(PLANE6(ENROUTE)) = 15
GO TO LOOP4
ELSE
IF RUNWAY(ALTERNATE.BASE(PLANE6(ENROUTE))) IS NOT EQUAL TO CLOSED
LET DESTINATION.BASE(PLANE6(ENROUTE)) =
  ALTERNATE.BASE(PLANE6(ENROUTE))
GO TO LOOP4
ELSE
LET X1 = X.POS(DESTINATION.BASE(PLANE6(ENROUTE)))
LET Y1 = Y.POS(DESTINATION.BASE(PLANE6(ENROUTE)))
CALL CLOSEST.BASE GIVEN X1 AND Y1
  YIELDING NEAREST.BASE AND MIN.DISTANCE
LET DESTINATION.BASE(PLANE6(ENROUTE)) = NEAREST.BASE
'LOOP4'
LOOP
RETURN
END
```

STRATEGY 2 MODULE

```
ROUTINE TO.DECISION.RULES GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 IF (MSN.TYPE(PLANE)=3 OR MSN.TYPE(PLANE)=5) AND
    RUNWAY(DESTINATION.BASE(PLANE)) = CLOSED
 GO CHECK.NEAREST.SUPPORT.BASE
 ELSE
 IF (MSN.TYPE(PLANE)=2 OR MSN.TYPE(PLANE)=4) AND
      RUNWAY(DESTINATION.BASE(PLANE)) = CLOSED
 LET MSN.TYPE(PLANE) = 1
 ELSE REGARDLESS
 GO RETURN
 'CHECK.NEAREST.SUPPORT.BASE'
 CALL NRST.SUPPORT.BASE GIVEN PLANE,
   X.LAST.POS(PLANE) AND Y.LAST.POS(PLANE)
 'RETURN'
RETURN
END

ROUTINE DEP.PT.DECISION.RULES GIVEN PLANE
 DEFINE PLANE AS AN INTEGER VARIABLE
 LET X1 = X.LAST.POS(PLANE)
 LET Y1 = Y.LAST.POS(PLANE)
 LET X2 = X.POS(DESTINATION.BASE(PLANE))
 LET Y2 = Y.POS(DESTINATION.BASE(PLANE))
 LET D = DISTANCE(X1,X2,Y1,Y2)
 LET MISSION.TIME.REMAINING(PLANE) = MISSION.TIME.REMAINING(PLANE)
    - D/1440.
 LET TOTAL.FUEL.REMAINING.TIME(PLANE)=TOTAL.FUEL.REMAINING.TIME(PLANE)
    - D/1440.
 SCHEDULE AN ARRIVAL.AT.IAF GIVEN PLANE IN D MINUTES
RETURN
END

ROUTINE AB.DECISION.RULES
 DEFINE FINAL.AREA AS AN INTEGER VARIABLE
 DEFINE NEAREST.BASE AS AN INTEGER VARIABLE
 PRINT 1 LINE AS FOLLOWS
LANDINGS AT CLOSED BASE
 FOR EACH LANDING IN EV.S(I.LANDING)
 WITH RUNWAY(CRNT.BASE(PLANE4(LANDING))) = CLOSED
 DO THE FOLLOWING
 IF (MSN.TYPE(PLANE4(LANDING))=2 OR MSN.TYPE(PLANE4(LANDING))=4) AND
 MSN.STAGE(PLANE4(LANDING))=OTHER.BASE.PATTERN AND
```

```
RUNWAY(HOME.BASE(PLANE4(LANDING))) IS NOT EQUAL TO CLOSED
LET DESTINATION.BASE(PLANE4(LANDING))=HOME.BASE(PLANE4(LANDING))
ELSE
CALL NRST.SUPPORT.BASE GIVEN PLANE4(LANDING),
   X.LAST.POS(PLANE4(LANDING)) AND Y.LAST.POS(PLANE4(LANDING))
REGARDLESS
LOOP
FOR EACH DEPARTURE.POINT IN EV.S(I.DEPARTURE.POINT)
WITH RUNWAY(CRNT.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
AND MSN.STAGE(PLANE2(DEPARTURE.POINT)) IS NOT EQUAL TO JUST.TOOKOFF
DO THE FOLLOWING
LET EMER.DIVERT.FLAG(PLANE2(DEPARTURE.POINT)) = DIVERT
IF (MSN.TYPE(PLANE2(DEPARTURE.POINT))=2 OR
MSN.TYPE(PLANE2(DEPARTURE.POINT))=4) AND
MSN.STAGE(PLANE2(DEPARTURE.POINT))=OTHER.BASE.PATTERN   AND
RUNWAY(HOME.BASE(PLANE2(DEPARTURE.POINT))) IS NOT EQUAL TO CLOSED
LET DESTINATION.BASE(PLANE2(DEPARTURE.POINT)) =
   HOME.BASE(PLANE2(DEPARTURE.POINT))
ELSE
CALL NRST.SUPPORT.BASE GIVEN PLANE2(DEPARTURE.POINT),
   X.LAST.POS(PLANE2(DEPARTURE.POINT)) AND
   Y.LAST.POS(PLANE2(DEPARTURE.POINT))
REGARDLESS
LOOP
FOR EACH ARRIVAL.AT.IAF IN EV.S(I.ARRIVAL.AT.IAF)
DO THE FOLLOWING
IF (MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 2 OR
   MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 4) AND
   RUNWAY(HOME.BASE(PLANE3(ARRIVAL.AT.IAF))) = CLOSED
LET MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 3
ELSE REGARDLESS
IF (MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 2 OR
   MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 4) AND
   RUNWAY(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF))) = CLOSED
LET DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)) =
   HOME.BASE(PLANE3(ARRIVAL.AT.IAF))
LET MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 1
ELSE REGARDLESS
IF (MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 1 OR
   MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 3 OR
   MSN.TYPE(PLANE3(ARRIVAL.AT.IAF)) = 5) AND
   RUNWAY(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF))) = CLOSED
LET D1 = (TIME.V-LKT(PLANE3(ARRIVAL.AT.IAF))) * 1440
LET X1 = X.LAST.POS(PLANE3(ARRIVAL.AT.IAF))
LET Y1 = Y.LAST.POS(PLANE3(ARRIVAL.AT.IAF))
LET X2 = X.POS(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)))
LET Y2 = Y.POS(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)))
LET D = DISTANCE(X1,X2,Y1,Y2)
```

```
  IF D1 = 0
  LET X.CURRENT.POS = X.LAST.POS(PLANE3(ARRIVAL.AT.IAF))
  LET Y.CURRENT.POS = Y.LAST.POS(PLANE3(ARRIVAL.AT.IAF))
  LET D2 = D
  GO TO PRNT
  ELSE
  CALL POSITION GIVEN X1,X2,Y1,Y2,D1 AND D
  YIELDING X.CURRENT.POS,Y.CURRENT.POS AND D2
'PRNT'
LET X.LAST.POS(PLANE3(ARRIVAL.AT.IAF)) = X.CURRENT.POS
LET Y.LAST.POS(PLANE3(ARRIVAL.AT.IAF)) = Y.CURRENT.POS
LET LKT(PLANE3(ARRIVAL.AT.IAF)) = TIME.V
LET X1 = X.CURRENT.POS
LET Y1 = Y.CURRENT.POS
CALL NRST.SUPPORT.BASE GIVEN PLANE3(ARRIVAL.AT.IAF),
  X1 AND Y1
LET X2 = X.POS(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)))
LET Y2 = Y.POS(DESTINATION.BASE(PLANE3(ARRIVAL.AT.IAF)))
LET D = DISTANCE(X1,X2,Y1,Y2)
LET MISSION.TIME.REMAINING(PLANE3(ARRIVAL.AT.IAF)) =
  MISSION.TIME.REMAINING(PLANE3(ARRIVAL.AT.IAF)) - D/1440.
LET TOTAL.FUEL.REMAINING.TIME(PLANE3(ARRIVAL.AT.IAF)) =
  TOTAL.FUEL.REMAINING.TIME(PLANE3(ARRIVAL.AT.IAF)) - D/1440.
CANCEL THE ARRIVAL.AT.IAF
RESCHEDULE THE ARRIVAL.AT.IAF GIVEN PLANE3(ARRIVAL.AT.IAF) IN D MINUTES
ELSE REGARDLESS
LOOP
FOR EACH LANDING IN EV.S(I.LANDING)
DO THE FOLLOWING
IF (MSN.TYPE(PLANE4(LANDING))=2 OR MSN.TYPE(PLANE4(LANDING))=4) AND
  RUNWAY(HOME.BASE(PLANE4(LANDING))) = CLOSED
LET MSN.TYPE(PLANE4(LANDING)) = 3
SUBTRACT 1 FROM TRANSIENT.PARKING.SPACES(CRNT.BASE(PLANE4(LANDING)))
ELSE REGARDLESS
LOOP
FOR EACH DEPARTURE.POINT IN EV.S(I.DEPARTURE.POINT)
DO THE FOLLOWING
IF (MSN.TYPE(PLANE2(DEPARTURE.POINT))=2 OR
  MSN.TYPE(PLANE2(DEPARTURE.POINT))=4) AND
  RUNWAY(HOME.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 3
SUBTRACT 1 FROM
  TRANSIENT.PARKING.SPACES(CRNT.BASE(PLANE2(DEPARTURE.POINT)))
ELSE REGARDLESS
LOOP
FOR EACH ENROUTE IN EV.S(I.ENROUTE)
DO THE FOLLOWING
IF MSN.TYPE(PLANE6(ENROUTE)) = 2
```

```
AND RUNWAY(DESTINATION.BASE(PLANE6(ENRCUTE))) = CLOSED
LET MSN.TYPE(PLANE6(ENROUTE)) = 1
LET DESTINATION.BASE(PLANE6(ENROUTE)) =
   HOME.BASE(PLANE6(ENROUTE))
ELSE REGARDLESS
IF MSN.TYPE(PLANE6(ENROUTE)) = 2
AND RUNWAY(HOME.BASE(PLANE6(ENROUTE))) = CLOSED
LET MSN.TYPE(PLANE6(ENROUTE)) = 5
ELSE REGARDLESS
IF MSN.STAGE(PLANE6(ENROUTE)) = MSN.COMPLETE AND
   RUNWAY(DESTINATION.BASE(PLANE6(ENROUTE))) = CLOSED
CALL NRST.SUPPORT.BASE GIVEN PLANE6(ENROUTE),
   X.LAST.POS(PLANE6(ENROUTE)) AND Y.LAST.POS(PLANE6(ENROUTE))
GO TO LOOP1
ELSE
IF 1.MSN.OPTION(PLANE6(ENROUTE)) = 0 AND
    2.MSN.OPTION(PLANE6(ENROUTE)) = 0 AND
    3.MSN.OPTION(PLANE6(ENROUTE)) > 0  AND
   RUNWAY(DESTINATION.BASE(PLANE6(ENROUTE))) = CLOSED
LET X1 = X.COOR(MSN.3.AREA(PLANE6(ENROUTE)))
LET Y1 = Y.COOR(MSN.3.AREA(PLANE6(ENROUTE)))
CALL NRST.SUPPORT.BASE GIVEN PLANE6(ENROUTE),X1 AND Y1
ELSE REGARDLESS
IF 1.MSN.OPTION(PLANE6(ENROUTE)) = 0 AND
    2.MSN.OPTION(PLANE6(ENROUTE)) > 0 AND
    3.MSN.OPTION(PLANE6(ENROUTE)) = 0  AND
   RUNWAY(DESTINATION.BASE(PLANE6(ENROUTE))) = CLOSED
LET X1 = X.COOR(MSN.2.AREA(PLANE6(ENROUTE)))
LET Y1 = Y.COOR(MSN.2.AREA(PLANE6(ENROUTE)))
CALL NRST.SUPPORT.BASE GIVEN PLANE6(ENROUTE),X1 AND Y1
ELSE REGARDLESS
IF 1.MSN.OPTION(PLANE6(ENROUTE)) = 0 AND
    2.MSN.OPTION(PLANE6(ENROUTE)) > 0 AND
    3.MSN.OPTION(PLANE6(ENROUTE)) > 0  AND
   RUNWAY(DESTINATION.BASE(PLANE6(ENROUTE))) = CLOSED
LET X1 = X.COOR(MSN.3.AREA(PLANE6(ENROUTE)))
LET Y1 = Y.COOR(MSN.3.AREA(PLANE6(ENRCUTE)))
CALL NRST.SUPPORT.BASE GIVEN PLANE6(ENROUTE),X1 AND Y1
ELSE REGARDLESS
'LOOP1'
LOOP
FOR EACH .MISSION IN EV.S(I..MISSION)
DO THE FOLLOWING
IF MSN.TYPE(PLANE5(.MISSION)) = 2 AND
   RUNWAY(DESTINATION.BASE(PLANE5(.MISSION))) = CLOSED
LET MSN.TYPE(PLANE5(.MISSION)) = 1
LET DESTINATION.BASE(PLANE5(.MISSION)) =
   HOME.BASE(PLANE5(.MISSION))
```

```
ELSE REGARDLESS
IF MSN.TYPE(PLANE5(.MISSION)) = 2 AND
  RUNWAY(HOME.BASE(PLANE5(.MISSION))) = CLOSED
LET MSN.TYPE(PLANE5(.MISSION)) = 5
ELSE REGARDLESS
IF 3.MSN.OPTION(PLANE5(.MISSION)) > 0 AND
  RUNWAY(DESTINATION.BASE(PLANE5(.MISSION))) = CLOSED
LET X1 = X.COOR(MSN.3.AREA(PLANE5(.MISSION)))
LET Y1 = Y.COOR(MSN.3.AREA(PLANE5(.MISSION)))
CALL NRST.SUPPORT.BASE GIVEN PLANE5(.MISSION),X1 AND Y1
ELSE REGARDLESS
IF 2.MSN.OPTION(PLANE5(.MISSION)) > 0 AND
  RUNWAY(DESTINATION.BASE(PLANE5(.MISSION))) = CLOSED
LET X1 = X.COOR(MSN.2.AREA(PLANE5(.MISSION)))
LET Y1 = Y.COOR(MSN.2.AREA(PLANE5(.MISSION)))
CALL NRST.SUPPORT.BASE GIVEN PLANE5(.MISSION),X1 AND Y1
ELSE REGARDLESS
IF 1.MSN.OPTION(PLANE5(.MISSION)) > 0 AND
  RUNWAY(DESTINATION.BASE(PLANE5(.MISSION))) = CLOSED
LET X1 = X.COOR(MSN.1.AREA(PLANE5(.MISSION)))
LET Y1 = Y.COOR(MSN.1.AREA(PLANE5(.MISSION)))
CALL NRST.SUPPORT.BASE GIVEN PLANE5(.MISSION),X1 AND Y1
ELSE REGARDLESS
LOOP
FOR EACH DEPARTURE.POINT IN EV.S(I.DEPARTURE.POINT)
DO THE FOLLOWING
IF (MSN.TYPE(PLANE2(DEPARTURE.POINT))=3 OR
  MSN.TYPE(PLANE2(DEPARTURE.POINT))=4)
GO TO PROCESS
ELSE
IF 3.MSN.OPTION(PLANE2(DEPARTURE.POINT)) > 0
LET FINAL.AREA = MSN.3.AREA(PLANE2(DEPARTURE.POINT))
ELSE
  IF 2.MSN.OPTION(PLANE2(DEPARTURE.POINT)) > 0
  LET FINAL.AREA = MSN.2.AREA(PLANE2(DEPARTURE.POINT))
  ELSE
  LET FINAL.AREA = MSN.1.AREA(PLANE2(DEPARTURE.POINT))
  REGARDLESS
REGARDLESS
'PROCESS'
IF MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 2 AND
  RUNWAY(DESTINATION.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 1
ELSE REGARDLESS
IF MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 2 AND
  RUNWAY(HOME.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 5
ELSE REGARDLESS
```

```
IF MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 4 AND
  RUNWAY(HOME.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 3
ELSE REGARDLESS
IF (MSN.TYPE(PLANE2(DEPARTURE.POINT))=1 OR
  MSN.TYPE(PLANE2(DEPARTURE.POINT))=5) AND
  RUNWAY(DESTINATION.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET X1 = X.COOR(FINAL.AREA)
LET Y1 = Y.COOR(FINAL.AREA)
CALL NRST.SUPPORT.BASE GIVEN PLANE2(DEPARTURE.POINT),X1 AND Y1
ELSE REGARDLESS
IF MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 3 AND
RUNWAY(DESTINATION.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET X1 = X.POS(DESTINATION.BASE(PLANE2(DEPARTURE.POINT)))
LET Y1 = Y.POS(DESTINATION.BASE(PLANE2(DEPARTURE.POINT)))
CALL NRST.SUPPORT.BASE GIVEN PLANE2(DEPARTURE.POINT),X1 AND Y1
ELSE REGARDLESS
IF MSN.TYPE(PLANE2(DEPARTURE.POINT)) = 4 AND
  RUNWAY(DESTINATION.BASE(PLANE2(DEPARTURE.POINT))) = CLOSED
LET X1 = X.LAST.POS(PLANE2(DEPARTURE.POINT))
LET Y1 = Y.LAST.POS(PLANE2(DEPARTURE.POINT))
CALL CLOSEST.BASE GIVEN X1 AND Y1
  YIELDING NEAREST.BASE AND MIN.DISTANCE
LET DESTINATION.BASE(PLANE2(DEPARTURE.POINT)) = NEAREST.BASE
ELSE REGARDLESS
LOOP
RETURN
END
```

BIBLIOGRAPHY

1. Kiviat, P. J., Villanueva, R., and Markowitz, H. M., SIMSCRIPT II-5 Programming Language, 2d ed., Consolidated Analysis Centers, Inc., 1975.

2. <u>SIMSCRIPT II.5 Reference Handbook</u>, Consolidated Analysis Centers, Inc., 1976.

3. <u>User's Manual</u>, 2d ed., W. R. Church Computer Center, Naval Postgraduate School, Monterey, California, 1974.

4. Minutes of General Officers' Panel on Mid-Air Collision Potential, August 1975 - January 1976, chaired by Major General Billy J. Ellis, HQ USAF/XOO.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 39 Naval Postgraduate School Monterey, California 93940	1
4.	LT Ellen F. Roland, USN, Code 55Ao Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
5.	Assoc. Professor A. F. Andrus, Code 55As Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6.	Major John T. Malokas, Jr., USAF 795 East 232 Street Euclid, Ohio 44123	1
7.	Captain Arvid Paul Pederson, USAF 22328 Capote Drive Salinas, California 93907	1